VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FACULTY OF INFORMATION TECHNOLOGY
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

# PHYSICALLY-BASED MODELING AND SIMULATION

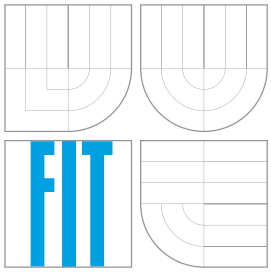
DISERTAČNÍ PRÁCE
PHD THESIS

AUTOR PRÁCE
AUTHOR

Ing. RADIM DVOŘÁK

BRNO 2014


BRNO UNIVERSITY OF TECHNOLOGY
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ


FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ


# FYZIKÁLNÍ MODELOVÁNÍ A SIMULACE

PHYSICALLY-BASED MODELING AND SIMULATION

DISERTAČNÍ PRÁCE
PHD THESIS


AUTOR PRÁCE — Ing. RADIM DVOŘÁK
AUTHOR

VEDOUCÍ PRÁCE — Doc. Ing. FRANTIŠEK ZBOŘIL, CSc.
SUPERVISOR



BRNO 2014

# Abstrakt

Disertační práce se zabývá modelováním znečištění ovzduší, jeho transportních a disperzních procesů ve spodní části atmosféry a zejména numerickými metodami, které slouží k řešení těchto modelů. Modelování znečištění ovzduší je velmi důležité pro předpověď kontaminace a pomáhá porozumět samotnému procesu a eliminaci následků. Hlavním tématem práce jsou metody pro řešení modelů popsaných parciálními diferenciálními rovnicemi, přesněji advekčně-difúzní rovnicí. Polovina práce je zaměřena na známou metodu přímek a je zde ukázáno, že tato metoda je vhodná k řešení určitých konkrétních problémů. Dále bylo navrženo a otestováno řešení paralelizace metody přímek, jež ukazuje, že metoda má velký potenciál pro akceleraci na současných grafických kartách a tím pádem i zvětšení přesnosti výpočtu. Druhá polovina práce se zabývá poměrně mladou metodou ELLAM a její aplikací pro řešení atmosférických advekčně-difúzních rovnic. Byla otestována konkrétní forma metody ELLAM společně s navrženými adaptacemi. Z výsledků je zřejmé, že v mnoha případech ELLAM překonává současné používané metody.



# Abstract

The thesis deals with the modeling of air pollution transportation and dispersion processes in the atmosphere, more precisely with the numerical approaches to solve such models. The modeling of air pollution has a great importance for prediction of the contaminations and it helps with understanding of the process and with elimination of its consequences. The models which are described by partial differential equations, namely advection-diffusion equations, and thus they can be solved by numerous analytical/numerical methods are in the scope of the thesis. In particular, well known method of lines (MoL) and several models based on it together with the possibility to accelerate the computation are studied in the first half of the work. It is shown that MoL approach is still suitable for many concrete models and it has a great potential for parallelization on graphics cards. Quite young ELLAM method and its application to solved atmospheric advection-diffusion equations is the second objective. A concrete form of ELLAM method and its proposed adaptation approaches are evaluated and it is shown that it overcomes the current state of the art methods in many cases.



# Klíčová slova

Modelování fyzikálních jevů, modelování znečištění ovzduší, parciální diferenciální rovnice, advekčně-difúzní rovnice, numerické metody, ELLAM

# Keywords

Physically-based modeling, modeling of air pollution, partial differential equations, advection-diffusion equation, numerical methods, ELLAM

# Citation

Radim Dvořák: Physically-based Modeling and Simulation, PhD thesis, Brno, BUT FIT, 2014

# Physically-based Modeling and Simulation

## Declaration

I hereby declare that this thesis is my genuine work, created under the guidance of my supervisor Assoc. Prof. František Zbořil, CSc. All information sources and publications used are properly cited.

.....
Radim Dvořák
June 24, 2014


## Acknowledgment

I would like to thank my supervisor Assoc. Prof. František Zbořil and Assoc. Prof. Martin Drahanský for their guidance and valuable comments to my research work. I also would like to thank my friends, my colleagues at Faculty of Information Technology, Brno University of Technology, and especially to my parents for their support.

# Contents

# List of Figures

# List of Tables

# Nomenclature

$C$ concentration of pollutant (kg $m^{-3}$). 4, 14–17, 20, 23, 25, 41, 42, 48, 57–61, 65–69, 71, 81–83, 91

$D$ diffusion coefficient ($m^2$ $s^{-1}$). 16, 17, 20, 22, 23, 25, 38, 39, 45, 50–53, 57–61, 65–68, 75, 81

$H$ mixing atmospheric height (m) - the top domain boundary above the ground. 14, 15, 21, 22, 53, 59, 65, 66, 79–81

$K$ von Karmann constant (dimensionless). 21, 22, 53, 79, 81

$L$ Monin–Obukhov length (m) - describes the effects of buoyancy on turbulent flows. 20–22, 79, 81

$Q_a$ area emission rate of the pollution source (kg $m^{-2}$ $s^{-1}$). 15

$Q$ emission rate of the pollution source (kg $s^{-1}$). 14, 58–60, 65, 66, 76, 81

$R$ mixing ratio of pollutant in atmosphere (kg $kg^{-1}$). 48, 49, 52, 53

$S$ source term (kg $s^{-1}$). 38, 52, 53, 67, 68

$W$ gravitational settling velocity (m $s^{-1}$). 20, 23, 57–62, 76

$\rho$ air density (kg $m^{-3}$). 48, 49, 52, 53

$\sigma_y$ standard deviation of horizontal distribution of plume concentration (m); evaluated at the downwind distance and for the appropriate stability. 14, 15, 18, 19

$\sigma_z$ standard deviation of vertical distribution of plume concentration (m); evaluated at the downwind distance and for the appropriate stability. 14, 15, 18, 19

$a_\star$ friction velocity (m $s^{-1}$). 20–22, 53, 79, 81

$a$ wind velocity, advection coefficient (m $s^{-1}$). 14, 15, 17, 20–22, 35, 36, 38, 39, 43, 44, 48–53, 57–62, 65–68, 76, 86

$h$ effective height of emission (m). 15

$v$ deposition velocity (m $s^{-1}$). 58–60, 76

$w_\star$ convective velocity scale (m $s^{-1}$). 22, 79, 81

$z_0$ surface roughness length (m). 20, 21, 79

**ADE** advection-diffusion equation. 10, 14, 30, 33, 34, 38, 39, 42, 45, 50, 52, 54–58, 63, 67, 93

**CFL** Courant–Friedrichs–Lewy number (dimensionless). 39

**COR** correlation coefficient (dimensionless). 42, 82, 90

**DistrE** mass distribution ratio error (dimensionless). 41, 84, 85

**ELLAM** Eulerian-Lagrangian localized adjoint method. 3, 4, 6, 10, 11, 50–52, 55, 57, 67, 69–73, 75, 83–93

**FA2** fractional of data (dimensionless). 42, 82, 90

**FB** fractional bias (dimensionless). 42, 82, 90

**FDM** finite difference method. 24–26, 29, 30

**FEM** finite element method. 24, 25, 29, 30, 37, 38, 50, 69

**FS** fractional standard deviation (dimensionless). 42, 82, 90

**FVM** finite volume method. 30

**MAE** mean absolute error (dimensionless). 40

**MassE** mass error (dimensionless). 6, 40, 84, 85, 88, 89

**MoL** method of line. 10, 24–26, 57, 58, 60, 63, 65, 66, 72, 76, 80–82, 90, 92, 93

**MSE** mean squared error (dimensionless). 40

**NMSE** normalized mean square error (dimensionless). 41, 82, 90

**ODE** ordinary differential equation. 26, 28, 34, 58–60, 77, 79

**PDE** partial differential equation. 24–26, 29, 30, 34, 42, 45, 46, 55, 65

**PeakE** peak error (dimensionless). 41, 84, 85

**RMSE** relative root mean squared error (dimensionless). 41, 84, 85

**WENO** weighted essentially non-oscillatory method. 3, 44–48, 55, 93

# Chapter 1

# Introduction

The field of physically-based modeling and simulation includes a large variety of problems. It may be modeling of microscopic world on atom level or modeling of galaxies collision. Although it covers a large amount of problems it is in the most cases possible to describe them by ordinary or partial differential equations. The differences are primarily in scale, sizes and dynamics of the equation coefficients.

The thesis deals with the modeling of air pollution dispersion in the atmosphere, more precisely with the numerical approaches to solve such models. The modeling of air pollution has a great importance for prediction of the contaminations and it helps with understanding of the process and with elimination of its consequences. The very important in the latter case is to know what space and what concentrations the species could reach and thus to be able to better remove or prevent damages.

The history of these kinds of models is dated to the $19^{th}$ century when Reynolds formulated a criterion for the change of laminar to turbulent flow [7]. Since then the diffusion phenomena has started to be studied and on the edge of the $19^{th}$ and $20^{th}$ centuries the first aerosol deposition models were described. Later on, the models with chemical reactions appeared because chemical reactions influence the amount of pollutant concentration.

There are two approaches for the solution of the atmospheric equations being developed. The first one is the analytical solution that is applied to the simpler cases where the certain parameters or attributes can be omitted. These techniques can serve as a validation for the numerical methods dealing with the more complex problems. Many numerical approaches do exist and they have been developing extensively, particularly in last decades. However, there is still much to enhance. The more precise and the more faster the calculation is the more complex problems can be solved in more details. Therefore, the theme of this thesis is the accurate and fast solution of the models of atmospheric pollutant dispersion.

## 1.1 Motivation and goals

The problem of air pollutant modeling is very large and cannot be expressed only by one single equation. Therefore, it is needed to describe the exact domain and on its basis to highlight its goals. The two main viewpoints should be taken into account when categorizing the model - a scale and the pollution processes.

Air pollution does exist at all scales, from extremely local to very global ones. The scales can be categorized into several areas: local, urban, regional, continental and global. The range of influence of the pollution can be from molecular level (e.g. nanoparticles) to

entire planetary (e.g. greenhouse gases diffusion in troposphere). One of the main goals of the thesis is to predict pollution made locally caused by accidents or smaller local pollutant emitters. The scale of such problems are within several kilometers, typically up to 5 km.

There are two major aspects/processes that influence the air pollutant model design - transportation and transformation. The first one refers to the processes which influence substance movement through the environment. The transformation process changes the type of substance from one to another.

The scope of the thesis is local scale and transportation process. Many models still exist in this scope one can select from. The thesis deals with models which are described by partial differential equations, more precisely by advection-diffusion equations (ADE), and thus they can be solved by many analytical/numerical methods. As the computational power of modern computers increases dramatically in the past decades and years, numerous numerical methods were developed and tested in order to solve complex air pollution models and also the thesis focuses on them. The well known method of lines is in the scope of the first half of the thesis, especially its computational parallel design in order to use current multi-core central processing unit (CPU) and graphics processing unit (GPU) computational power and fasten its computational process. Quite young and promising ELLAM method [10] is studied in the second half of the thesis and it is compared with the state of the art methods for pure advection problems. At the end, the evaluation of the method for real advection-diffusion models is also presented.

The main goals of the dissertation thesis are:

1. To design a way of parallelization of the method of lines (MoL);
2. To enhance numerical approach (ELLAM) to solve ADE models.

The mentioned goals are addressed in chapter 4 and among others, the results of the contributions are presented in chapter 5. In particular, the parallel version of method of lines and its experiments are described in sections 4.1.3 and 5.1.2.

The concrete ELLAM method improvements are suggested and described in section 4.2.2 and the experiments and their results are presented in section 5.2.

## 1.2 Thesis structure

The basic terms of pollutant dispersion modeling in the atmospheric planetary boundary layer are described in chapter 2. The most often used approaches are outlined there and the concrete models used further in the thesis are presented in more details.

The chapter 3 includes the overview of the mostly used numerical approaches to solve partial differential equations in general. Further, the basic terms and properties of advection-diffusion equations are presented and on their basis the specialized numerical methods are described in detail. All the methods presented in the chapter are tested and compared with each other using simple one-dimensional experiments. The goal of the chapter is also to choose the existing methods for further investigations.

The concrete chosen methods and suggestions for their improvements are presented in chapter 4. Here the relatively simple methods for various forms of advection-diffusion equation based on method of lines framework are presented. Also its parallel version is also described here. The second part of the chapter is dedicated to ELLAM method, the form which was implemented in this work and the suggestions for its improvements together with simple one-dimensional tests.

The designed methods were tested in relatively big amount of experiments. The results of the performed experiments are presented in chapter 5 for method of lines, its parallel version and for ELLAM method and its modified version using both artificial and real models.

The final conclusion and possible future work are outlined in the last chapter 6.

# Chapter 2

# Pollutant dispersion modeling

This chapter deals with the problematic of dispersion modeling. First, the definition of air pollution itself is stated including important scale factor, basic concepts description and definition of terms. In the second section, concrete model and its parts used further in the text are presented in detail.

## 2.1 Air pollution

If one thinks about pollution it is important to answer the question what amount of matter in the air is just impurity and what amount shall be considered as pollutant.

### 2.1.1 Definition

The answer on the above stated question is not straightforward but it can be shown that the awareness of several aspects can reveal it quite well. First, the context of pollutant is a very important factor. When pollutant reaches the receptor one shall ask what is physical, chemical and biological nature of the receiver (e.g. person, species or the entire population), what is the health condition of the receiver, what is the composition of the pollutant etc. The answers to such questions have to be known in order to state the degree of harm.

The further viewpoint is related to response of the receiver to the pollutant. In case of air pollution it holds in most cases that the more poison to which one is exposed the greater the harm. The above statements can lead us to the following definition [57] - *The presence of contaminants or pollutant substances in the air that interfere with human health or welfare, or produce other harmful environmental effects.*

### 2.1.2 Scale factor

Air pollution exists at all scales, from extremely local to very global ones. The scales can be categorized into several areas: local, urban, regional, continental and global. The range of influence of the pollution can be from molecular level (e.g. nanoparticles) to entire planetary (e.g. greenhouse gases diffusion in troposphere). The local scale is up to about 5 km of the earth's surface. The urban scale extends to the order of 50 km. The regional scale is from 50 to 500 km. Continental scales are from 500 to several 1000 km. The global scale extends worldwide.

The scope of this thesis deals with scale factor of local/partially urban categories. These air pollution problems are usually characterized by one or several large emitters or a large

number of relatively small emitters. The lower the release height of a source, the larger the potential impact for a given release might be.

The examples are local emitters of carbon monoxide from motor vehicles which leads to local relatively high pollution near roadways. Local pollution caused by sudden accident near the industrial facilities can be another example of local pollutions. In these cases, the concentration of pollutant is highest near its source and quite rapidly diminishes with distance. This phenomenon is known as *concentration gradient*.

The higher positioned sources can also lead to local pollutions. It can happen especially when unstable meteorological conditions tend to lower the contaminant to the ground with high concentrations. The examples could be smaller power plants or local industrial facilities.

### 2.1.3 Basic terms and concepts

There are several conceptual terms needed to be considered before the details of model are described. These are sources/sinks of pollution, receptors, transport and dispersion.

#### Sources and sinks

The places pollutants are emitted from are called *sources*. The sources can be of artificial or natural manner. The artificial ones include gas pollutions from industry, vehicles and other facilities built by human. The natural sources can be the respirations from plants, animals and fallout of what was once living matter. Other natural sources include volcanoes and naturally caused forest fires. The pollutants disappear in places called *sinks*. These are soil, vegetation and water areas such as oceans.

#### Receptors

A *receptor* can be the plant or animal that is affected by a pollutant. The interface between a pollutant and a receptor can be its surface (e.g. skin) or its part, lungs when the pollutant is inspired by animal breathing or when its eyes are irritated. Also a material can be the receptor - paper, leather, clothes, etc. Some artificial receptors are made to measure the concentration of the pollution in specific places. These can be used either for pollution statistical measurement or for further processing such as future prediction.

#### Transport and dispersion

A *transport* is the process that moves the pollution from the source to the receptor. The simplest examples of the source-receptor system is the point source and single receptor tuple such as a chimney and a building which is 5 km far away. The pollutant flows directly to the receptor when wind blows from source to receptor along the line connecting the two points and when its direction is from the source to the receptor. The receptor is affected by a pollutant, however, the matter does not form the same circular shape all the way it passes. On the contrary, the plume particles move from the edges to the surrounding air and the particles from surrounding air are moving inside plume due to turbulent eddies. Next, if wind speed is smaller/larger than emitting speed plume slows/accelerates and is deformed until it reaches the wind speed.

The two processes, mixing with surroundings and plume deforming (stretch-out), tend to alter the concentration of the pollutant less at the receptor than at the source. The

sum of these processes is called *diffusion*. However, the term diffusion has a substantially different meaning in chemistry. Substances diffuse according to Fick's law of diffusion [57], wherein the concentration diminishes with distance from the source. This is known as a concentration gradient. Therefore, *dispersion* is the preferred term.

#### Transformation

Next to the transportation one should consider also the process of *transformation* which refers to those processes that change a substance of interest into other substance. The two primary modes of transformation are physical (transformations caused by physical laws, such as radioactive decay) and chemical (transformations caused by chemical or biological reactions, such as dissolution and respiration) [52].

### 2.1.4 Modeling techniques

There exist many techniques and corresponding models describing air pollution phenomena. The following sections outline the basic modeling techniques while one in the scope of this thesis, gradient transport model, is described in more detail.

#### Gaussian plume model

Gaussian plume model is the solution of the ADE equation where wind and diffusion variables remain constant. It is used to predict pollution from continuously emitting point source such as industrial stack [13].

The equation includes three axes - downwind ($x$ axis), crosswind ($y$ axis) and vertical axis originating from the ground surface ($z$ axis). The model assumes that concentrations are proportional to the emission rate, that these concentrations are diluted by the wind at the point of emission at a rate inversely proportional to the wind speed, and that the time-averaged pollutant concentrations crosswind and vertically near the source are well described by Gaussian or normal (bell-shaped) distributions [57]. The standard deviations of the distributions $\sigma_y$ (m) and $\sigma_z$ (m) can be estimated from empirical values. The more details are stated in section 2.1.5.

The equations for the Gaussian plume solution follow. For stable atmospheric conditions the solution is of the form [57]

$$C(x,y,z) = Q\frac{1}{a_x}\left(\frac{g_1}{\sqrt{2\pi}\sigma_y}\right)\left(\frac{g_2}{\sqrt{2\pi}\sigma_z}\right), \tag{2.1}$$

where $C$ (kg m$^{-3}$) is pollutant concentration, $Q$ (kg s$^{-1}$) is emission rate and $a_x$ (m s$^{-1}$) is wind speed along $x$ axis. Lets further consider mixing height $H$ (m). For unstable or neutral conditions where $\sigma_z > 1.6H$

$$C(x,y,z) = Q\frac{1}{a_x}\frac{g_1}{\sqrt{2\pi}\sigma_y}\frac{1}{H}. \tag{2.2}$$

For unstable or neutral conditions where $\sigma_z < 1.6H$

$$C(x,y,z) = Q\frac{1}{a_x}\frac{g_1}{\sqrt{2\pi}\sigma_y}\frac{g_3}{\sqrt{2\pi}\sigma_z}. \tag{2.3}$$

For all equations 2.1, 2.2 and 2.3, $g_1$, $g_2$, $g_3$ are calculated as follows

$$
\begin{aligned}
g_1 &= \exp\left(\frac{-0.5y^2}{{\sigma_y}^2}\right) \\
g_2 &= \exp\left(\frac{-0.5(h-z)^2}{{\sigma_z}^2}\right) + \exp\left(\frac{-0.5(h+z)^2}{{\sigma_z}^2}\right) \\
g_3 &= \sum_{N=-\infty}^{\infty}\left(\exp\left(\frac{-0.5(h-z+2NH)^2}{{\sigma_z}^2}\right) + \exp\left(\frac{-0.5(h+z+2NH)^2}{{\sigma_z}^2}\right)\right),
\end{aligned}
\tag{2.4}
$$

where $h$ (m) is effective height of emission, in other words the height of the pollutant source. In calculation of $g_3$, putting $N = 4$ (summation from -4 to 4) is the sufficient approximation [57].

### Box model

The box model assumes a uniform mixing throughout the volume. It is useful for the first time approximation as a basis for further analysis. For steady-state emission and atmospheric conditions, with no upwind background concentrations, the concentration is given by [57]

$$
C = \frac{\Delta x Q_a}{Ha}, \tag{2.5}
$$

where $C$ (kg m$^{-3}$) is steady-state concentration, $\Delta x$ (m) is distance over which the emissions take place, $Q_a$ (kg m$^{-2}$ s$^{-1}$) is *area* emission rate, $H$ (m) is mixing height, and $a$ (m s$^{-1}$) is mean wind speed through the vertical extent of the box.

### Narrow plume hypothesis

By assuming that the principal contributors to the concentration at a receptor are the sources directly upwind, especially those nearby, the concentration due to area sources can be calculated using the vertical growth rate rather than uniform vertical mixing and considering the specific area emission rate of each area upwind of the receptor. Area emission rate changes in the crosswind direction are neglected as being relatively unimportant. The expansion in the vertical is usually considered using the Gaussian vertical growth [57].

### Trajectory models

Trajectory models actually move a vertical column with a square cross section. The column moves at the mean wind speed and the pollutants are added to the column bottom as the column moves over the space. The vertical dispersion is treated as direct mixing or using vertical mixing coefficient with a suitable profile [57].

The advantage of trajectory models is the ability to model chemical reactions and its minimal calculation requirements because concentrations can be calculated only for few locations of receptors. For simplicity of computation the column is remained to be vertical although due to different wind speed at its bottom and top it might be skewed. This is acceptable for urban application in the daytime, when winds are relatively uniform throughout the lower parts of atmosphere.

### Gradient transport models

There are defined certain physical variables in fluid mechanics which describe fluid behaviour in time and space. Fluid behaviour can be expressed as the change of its concentration in space and time. Concentration $C$ (kg m$^{-3}$) is defined as $C = M/V$, where $M$ (kg) is matter and $V$ (m$^{-3}$) is volume. The concentration change of fluid in atmosphere is characterized by two main processes - diffusion and advection.

Diffusion is the fluid property to randomly spread from places with higher concentrations to places with lower concentrations. The process is caused by random movement of molecules, so called Brownian motion. Because it is case of random process, the average case of the motion can be statistically described as follows [52].

Net flux of matter $f_x$ (kg s$^{-1}$) in one-dimensional case (along $x$ axis) is defined as a rate of matter amount change $k$ ($k \in [0, 1]$, with units of s$^{-1}$) in mutual places $l$ and $r$

$$f_x = k(M_l - M_r), \tag{2.6}$$

where $M_l$ (kg) and $M_r$ (kg) are mass weights in $l$ and $r$.

Lets consider the volume quantity $V = \Delta x \Delta y \Delta z$ and the concentrations at places $l$ and $r$ be defined as $C_l = M_l/(\Delta x \Delta y \Delta z)$ and $C_r = M_r/(\Delta x \Delta y \Delta z)$. Assuming one-dimensional case ($\Delta y \Delta z = 1$) and the finite difference approximation $\frac{\partial C}{\partial x} = \frac{C_r - C_l}{x_r - x_l}$, the following relation holds

$$\frac{\partial C}{\partial x} = \frac{C_r - C_l}{x_r - x_l} = \frac{M_r - M_l}{\Delta x (x_r - x_l)}. \tag{2.7}$$

Putting $\Delta x = (x_r - x_l)$ leads to the following diffusive flux expression $\vec{f_x}$ using concentrations

$$\vec{f_x} = -k(\Delta x)^2 \frac{\partial C}{\partial x} = -D_x \frac{\partial C}{\partial x}, \tag{2.8}$$

where $D_x$ (m$^2$ s$^{-1}$) is diffusion coefficient. Here, diffusive flux $\vec{f_x}$ is a vector quantity with units (kg m$^{-2}$ s$^{-1}$). To compute total mass flux rate in units (kg s$^{-1}$) the diffusive flux $\vec{f_x}$ has to be integrated over surface area. In one-dimensional case, it would mean $f_x = A\vec{f_x}$, where $A = \Delta y \Delta z$.

To describe the change in concentration of the diffusing mass over time at a point, the law of mass conservation is used. It says that the change of mass $M$ in volume $V$ over time $t$ is given by

$$\frac{\partial M}{\partial t} = \sum f_{in} - \sum f_{out}, \tag{2.9}$$

where $f_{in}$ and $f_{out}$ are diffusive mass fluxes in and out of the volume which could be described in $x$-direction by equation (2.8) as

$$\begin{aligned} \overline{f_{x,in}} &= -D_x \left.\frac{\partial C}{\partial x}\right|_{in}, \\ \overline{f_{x,out}} &= -D_x \left.\frac{\partial C}{\partial x}\right|_{out}. \end{aligned} \tag{2.10}$$

Multiplying fluxes in equation (2.10) by surface area $A = \Delta y \Delta z$ and substituting them to equation (2.9) result in net flux $f_x$ in $x$-direction

$$f_x = -D_x \Delta y \Delta z \left( \left. \frac{\partial C}{\partial x} \right|_{in} - \left. \frac{\partial C}{\partial x} \right|_{out} \right). \tag{2.11}$$

The expression $\left. \frac{\partial C}{\partial x} \right|_{out}$ can be approximated from $\left. \frac{\partial C}{\partial x} \right|_{in}$ using forward difference approximation from Taylor series expansion (see section 3.1.1 for the definition) as

$$\left. \frac{\partial C}{\partial x} \right|_{out} = \left. \frac{\partial C}{\partial x} \right|_{in} + \Delta x \frac{\partial}{\partial x} \left( \left. \frac{\partial C}{\partial x} \right|_{in} \right). \tag{2.12}$$

Substituting right hand side of equation (2.12) to equation (2.11) and omitting the subscript $_{in}$ give

$$f_x = -D_x \Delta x \Delta y \Delta z \left( \frac{\partial^2 C}{\partial x^2} \right). \tag{2.13}$$

After substituting $C$ ($M = C \Delta x \Delta y \Delta z$) and net flux $f_x$ into equation (2.9) we obtain the one-dimensional diffusion equation

$$\frac{\partial C}{\partial t} = D_x \frac{\partial^2 C}{\partial x^2}. \tag{2.14}$$

In case of advection, matter moves in one-dimensional space by certain speed $a_x$ (m $s^{-1}$). One-dimensional matter flux $f_{x,a}$ due to advection is defined as $f_{x,a} = a_x C$. Using mass conservation law and following the similar procedure as described for diffusive flux $f_{x,d}$, the advection equation will lead to the following form

$$\frac{\partial C}{\partial t} = -\frac{\partial a_x C}{\partial x}. \tag{2.15}$$

When both processes are taken into account they lead to basic one-dimensional advection-diffusion equation in the form

$$\frac{\partial C}{\partial t} + \frac{\partial a_x C}{\partial x} = D_x \frac{\partial^2 C}{\partial x^2}. \tag{2.16}$$

Extending 2.16 into three dimensions leads to the basic advection-diffusion equation in three-dimensional space with $x$ and $y$ and $z$ axes in the form

$$\frac{\partial C}{\partial t} + a_x \frac{\partial C}{\partial x} + a_y \frac{\partial C}{\partial y} + a_z \frac{\partial C}{\partial z} = D_x \frac{\partial^2 C}{\partial x^2} + D_y \frac{\partial^2 C}{\partial y^2} + D_z \frac{\partial^2 C}{\partial z^2}. \tag{2.17}$$

Equation (2.17) can be re-written using nabla operator $\nabla = \vec{x} \frac{\partial}{\partial x} + \vec{y} \frac{\partial}{\partial y} + \vec{z} \frac{\partial}{\partial z}$, where $\vec{x}$, $\vec{y}$, $\vec{z}$ are unit vectors along particular axes, to the form

$$\frac{\partial C}{\partial t} + \nabla \vec{a} C = \vec{D} \nabla^2 C, \tag{2.18}$$

where $\vec{a}$ is vector $(a_x, a_y, a_z)$ and $\vec{D}$ is vector $(D_x, D_y, D_z)$.

Finally, when diffusion coefficients are not constant, the general form of advection-diffusion equation used in gradient transport modeling will have the form

$$\frac{\partial C}{\partial t} + \nabla \vec{a} C = \nabla \left( \vec{D} \nabla C \right). \tag{2.19}$$

### 2.1.5 Dispersion parameters estimation

Proper input data has to be inserted into the model in order to make the simulation valid. The input data for physically-based models could be difficult to collect which is one of the main problems of air pollution models. The collecting of data is out of the scope of the thesis though, it is very important to know how it is done in order to properly evaluate the models and their solution. As was stated before, advection and diffusion/dispersion are the main factors and while wind velocity can be more or less accurately measured at site the dispersion coefficients cannot be. Therefore, several, simple to use, dispersion estimations were defined by empirical measurements and the actual values can be taken from data set according to observations of current meteorological conditions. In case of Gaussian plume and Gradient transport models, the standard deviations of the distributions can be used in place of diffusion/dispersion coefficients.

#### Direct measurements of wind fluctuations

The idea is to use the fluctuation statistics from fixed wind systems to estimate the dispersion over finite release of times. Direct measurement of turbulence, such as the standard deviation of fluctuations in the horizontal (azimuth) or vertical (elevation) components of wind direction ($\sigma_a$ or $\sigma_e$, respectively), is usually measured by a bivane or sonic anemometer [27].

To calculate $\sigma_y$ and $\sigma_z$ directly from fluctuation measurements, the following empirically derived equations can be used [57]

$$\sigma_y = x\sigma_a f_y, \tag{2.20}$$

$$\sigma_z = x\sigma_e f_z, \tag{2.21}$$

where $x$ (m) is downwind distance from the source and $f_y$, $f_z$ are functions of the form

$$f = \left(1 + 0.9\sqrt{\frac{t}{t_0}}\right)^{-1}, \tag{2.22}$$

where $t$ (s) is travel time, $t_0$ (s) is 1000 for $f_y$, $t_0$ is 500 for $f_z$ for unstable conditions, and $t_0$ is 50 for $f_z$ for stable conditions.

#### Classification of wind direction traces

Estimates of horizontal dispersion can be approximated from common wind direction traces in cases where special fluctuation data are missing. The atmosphere is classified into five categories $A$, $B_2$, $B_1$, $C$ or $D$. Category $A$ did not have enough data for proper estimation of dispersion parameters and therefore it was excluded from the data set. For other four classes, the coefficients and exponents for the dispersion parameters are given in table 2.1.

The horizontal $\sigma_y$ and vertical $\sigma_z$ spreads can be then calculated as

$$\sigma_y = ax^b \tag{2.23}$$

$$\sigma_z = cx^d \tag{2.24}$$

where $a$, $b$, $c$ and $d$ coefficients are taken from table 2.1 and $x$ (m) is distance from the source to the receptor.

| Type | a | b | c | d |
|---|---|---|---|---|
| $B_2$ | 0.40 | 0.91 | 0.41 | 0.91 |
| $B_1$ | 0.36 | 0.86 | 0.33 | 0.86 |
| $C$ | 0.32 | 0.78 | 0.22 | 0.78 |
| $D$ | 0.31 | 0.71 | 0.06 | 0.71 |

Table 2.1: Coefficients and exponents for Brookhaven Gustiness Classes [57].

| | Isolation | | | Night | |
|---|---|---|---|---|---|
| Surface wind speed ($ms^{-1}$) | Strong | Moderate | Slight | Thinly overcast or $\geq 4/8$ low cloud | $\leq 3/8$ cloud |
| <2 | A | A–B | B | — | — |
| 2–3 | A–B | B | C | E | F |
| 3–5 | B | B–C | C | D | E |
| 5–6 | C | C–D | D | D | D |
| >6 | C | D | D | D | D |

Table 2.2: Pasquill stability categories [57].

#### Classification of atmospheric stability

Pasquill [41] recommended the use of fluctuation measurements for dispersion estimates but also provided a scheme „for use in the likely absence of special measurements of wind structure“. In such cases there was clearly a need for broad estimates of dispersion „in terms of routine meteorological data“. The necessary parameters for the scheme consist of wind speed, insolation, and cloudiness, which are basically obtainable from routine observations and which are directly used to estimate Pasquill stability category using table 2.2.

Pasquill's dispersion parameters were restated into terms of $\sigma_y$ and $\sigma_z$ by Gifford [19] in order to use them directly in, e.g., Gaussian plume models. The parameters $\sigma_y$ and $\sigma_z$ are found by estimation from graphs [19]. For each Pasquill category, a function of the distance between source and receptor is defined. The values of functions mean directly the amount of the variances $\sigma_y$, $\sigma_z$, respectively. These parameter values are the most applicable for releases near the ground (within about 50 m). There exist other functions that estimate the parameters for larger problem scales [57] that are out of scope of this thesis.

## 2.2 Model description

A model requires two types of data inputs: information about the source or sources of pollution such as position, emission rate etc. and meteorological data such as wind velocity and turbulence. The model then simulates mathematically the transportation and dispersion of pollutant and certain models can also simulate physical or chemical transformation or even removal processes. The outputs of the model are the concentrations for particular locations of receptors in certain time frames.

The models further used in the thesis come from the basis model which includes advection, dispersion and deposition terms whose analytical solution was described by Ermak [18]

$$a_x \frac{\partial C}{\partial x} = \frac{\partial}{\partial x}\left(D_x \frac{\partial C}{\partial x}\right) + \frac{\partial}{\partial y}\left(D_y \frac{\partial C}{\partial y}\right) + W \frac{\partial C}{\partial z}. \tag{2.25}$$

The model (2.25) describes the solution of the problem with steady point source of pollutant with constant wind speed $a_x$ (m s$^{-1}$) along $x$ axis and with constant deposition velocity $W$ (m s$^{-1}$). Narrow terrain is taken into account only in the solution. The model, as well as other analytical solutions, can be used to validate and test the numerical solutions of more general and complex models. Currently, the analytical solutions include the simple models of wind dependent on height [32], diffusion coefficients changing with distance from source [49] or with non-stationary source [9].

By generalizing of equation (2.25) and by including a reaction term, advection-diffusion-reaction equation with deposition is formed in three-dimensional space as

$$\frac{\partial C}{\partial t} + \nabla \vec{a} C = \nabla\left(\vec{D} \nabla C\right) + W \frac{\partial C}{\partial z} \pm k C^n, \tag{2.26}$$

where $kC^n$ is reaction term described in subsection 2.2.4.

The terms in equation (2.26) can also be described by individual models which is the usual case in practical applications. The following subsections include their description with concrete examples.

### 2.2.1 Wind

One of the important wind effect is a dilution of the pollutant near the place of the source - the higher wind speed the more diluted pollutant. This has big impact on pollution process and solution of the simulation, therefore wind speeds used in estimating plume dispersion are generally estimated at the stack top. The second, more obvious, effect is that wind speed directly influences the travel time of the species.

There exist several wind models that describe the wind flow in atmosphere. One of the most commonly used models in past assumes constant wind direction and it uses power law profile. Wind speed $a(z)$ (m s$^{-1}$) at height $z$ (m) is then defined as

$$a(z) = a(z_r)\left(\frac{z}{z_r}\right)^{\alpha}, \tag{2.27}$$

where $a(z_r)$ (m s$^{-1}$) is measured wind speed at referenced height $z_r$ (m) and $\alpha$ is constant that depends on atmospheric stability and surface roughness length [32]. It should be noted that the power law wind profiles do not necessarily represent the data well and other models were designed since that time.

The other, also simple, model is one presented by Lin [32]

$$a(z) = a(z_r)\left(\frac{z}{z_r}\right)^{\alpha} - kz^{\alpha}, \quad k = \frac{a(z_r)}{z_r^{\alpha}}, \tag{2.28}$$

where $a(z_r)$ is measured wind speed at referenced height $z_r$ and $k$, $\alpha$ are constants that depend on atmospheric stability and surface roughness length [32].

Next wind model directly used in the experiments of this work is the one presented by Wortmann [60]. The special coefficients are needed in order to calculate the wind speed at a given position. These are Monin-Obukhov length $L$ (m), surface roughness length $z_0$ (m) and friction velocity $a_\star$ (m s$^{-1}$). Then, the wind velocity is given by

$$a(z) = a_\star \left[ \ln \frac{z}{z_0} - \Psi_m \frac{z}{L} + \Psi_m \frac{z_0}{L} \right], \quad z \leq z_b$$
$$a(z) = a(z_b), \quad z > z_b, \tag{2.29}$$

where $z_b = min(|L|, 0.1H)$ (m) with $H$ (m) defined as height of the unstable boundary layer.

The function $\Psi_m$ presented in equation (2.29) is of the form

$$\Psi_m = 2 \ln \left( \frac{1+A}{2} \right) + \ln \left( \frac{1+A^2}{2} \right) - 2 \tan^{-1}(A) + \frac{\pi}{2} \tag{2.30}$$

$$A = \left[ 1 - \frac{16z}{L} \right]^{0.25}. \tag{2.31}$$

The last wind model used in the experiments in this thesis is represented by equation of the form [56]

$$a(z) = \frac{a_\star}{K} \{ \ln \frac{z}{z_0} - \left[ 1 - 6.9 \frac{H}{L} \right] + \frac{z - z_0}{L} - \frac{6.9H}{2L} \left[ \frac{z^2}{H^2} - \frac{{z_0}^2}{H^2} \right] \}, \quad \text{where} \quad H/L > 0 \tag{2.32}$$

$$a(z) = \frac{a_\star}{K} \{ \ln \frac{z}{z_0} + \ln \left[ \frac{(1+\mu_0^2)(1+\mu_0)^2}{(1+\mu_0^2)(1+\mu_0)^2} \right] + 2[\tan^{-1}\mu - \tan^{-1}\mu_0] - \frac{2L}{33h}[\mu^3 - \mu_0^3] \}, \quad \text{where} \quad H/L < 0 \tag{2.33}$$

with

$$\mu = \left( 1 - 22 \frac{Hz}{LH} \right)^{0.25},$$
$$\mu_0 = \left( 1 - 22 \frac{Hz_0}{LH} \right)^{0.25}.$$

Here, $K$ is von Karman constant and the other variables have the same meaning as in equation (2.29).

All these models have been successfully used in wind velocity fields estimations and they were compared by authors with concrete field measurements under various conditions. Thus the models are further used in this thesis.

### 2.2.2 Dispersion

It was stated in section 2.1.5 that dispersion coefficients can be substituted by variances of the flux distribution in the simpler models. However, it turned out that for certain

conditions the variances are not enough and several works dealing with dispersion models appeared. The two representative models were chosen for this work since they need the same set of measured parameters as wind models presented in the previous section.

The first model is represented by [56]

$$\begin{aligned} D(z) &= Ka_\star H \frac{z}{H}\left(1-\frac{z}{H}\right)\left(1+9.2\frac{H}{L}\frac{z}{H}\right)^{-1} \\ &\qquad \text{where} \quad H/L > 0 \\ D(z) &= Ka_\star H \frac{z}{H}\left(1-\frac{z}{H}\right)\left(1-13\frac{H}{L}\frac{z}{H}\right)^{1/2} \\ &\qquad \text{where} \quad H/L < 0. \end{aligned} \tag{2.34}$$

The variables have the same meaning as in equation (2.29).

The vertical *eddy diffusivity* derived in [17] is the second example of diffusion model defined as

$$\begin{aligned} D(z) = w_\star H 0.22 &\left(\frac{z}{H}\right)^{1/3}\left(1-\frac{z}{H}\right)^{1/3} \\ &\left[1-\exp\left(-\frac{4z}{H}\right)-0.0003\exp\left(\frac{8z}{H}\right)\right] \end{aligned} \tag{2.35}$$

The variables have the same meaning as in equation 2.29 and $w_\star$ (m s$^{-1}$), convective velocity scale, is given by

$$w_\star = a_\star \sqrt[3]{\frac{-H}{KL}}.$$

Further in the thesis, the other dispersion model dependent on height $z$ (m) and downwind distance from source $x$ (m) is used [16]

$$D(x,z) = w_\star H \frac{0.38\psi^{2/3}X(z)\left[1+0.75\psi^{1/3}q^{-2/3}X(z)\right]}{\left[0.82q^{-1/3}+1.24\psi^{1/3}q^{-1}X(z)\right]^2}, \tag{2.36}$$

where $X(z) = (xw_\star)/(a(z)H)$ is non-dimensional distance defined by the ratio of travel time $x/a(z)$ (s) ($a(z)$ (m s$^{-1}$) is mean wind speed at given height $z$ (m)) and convective time-scale $H/w_\star$ (s), and

$$q = \left[1-\exp\left(-\frac{4z}{H}\right)-0.0003\exp\left(\frac{8z}{H}\right)\right].$$

The dissipation function $\psi$ has the following form

$$\psi^{1/3} = \left[\left(1-\frac{z}{H}\right)^2\left(\frac{z}{-L}\right)^{-2/3}+0.75\right]^{1/2}.$$

### 2.2.3 Deposition

Next to advection and diffusion terms, one could consider the drop and spread of fluid on terrain surface. This process is described by *deposition* model.

Deposition velocities of particles describe the change rate of matter on atmosphere/terrain surface interface and they are the functions of several variables including properties of terrain and particles and meteorological conditions. Sehmel and Hodgson [48] described the above terrain surface particle movement which depends on concentration $C$ (kg m$^{-3}$), vertical atmospheric diffusion $D_z$ (m$^2$ s$^{-1}$) and speed of gravitational sedimentation $W$ (m s$^{-1}$). Deposition flux $f_z$ (kg m$^{-2}$ s$^{-1}$) to a surface is described by [51]

$$f_z = D_z \frac{\partial C}{\partial z} + WC. \tag{2.37}$$

### 2.2.4 Reactions

For the complete list of terms used in the air pollution model equation, reaction term is outlined here. It describes the interaction of substance with the environment. Two cases can theoretically appear. Either the substance concentration decreases during the reaction process or increases. The example could be the decreasing of oxygen during burning or increasing of oxygen due to the photosynthetic process.

The two types of reactions are stated in general. Homogeneous reactions are placed in the whole space volume and they are described as a new term of advection-diffusion equation. On the other hand, heterogeneous reactions are present on space boundary and therefore it is described by a special boundary condition.

The change of reactant concentration $C$ (kg m$^{-3}$) of the species during time $t$ (s) is expressed by reaction coefficient $k$ (kg$^{-(n-1)}$ m$^{3(n-1)}$ s$^{-1}$) as [52]

$$\frac{\partial C}{\partial t} = R = \pm k C^n,$$

where $n$ is reaction order with respect to the given species. When the sign is positive the amount of species is rising and vice versa.

# Chapter 3

# Advection-diffusion equation and its solution

The concrete models used in the thesis are special cases of the general governing equation (2.26) which is a partial differential equation (PDE) consisting of derivatives of first and second order. PDE can be solved analytically where the solution is the exact one, or numerically where the solution is an approximation. Analytical solution is often hard or even impossible to find in many cases, especially for complex problems with complex boundary conditions.

In case of atmospheric equations, analytical solutions are not known in general, however, the solutions for special cases are derived and these can be used for validation of numerical methods which are then used for complex, more general models.

In the past century, many numerical methods have been developed for solving PDE of different kinds. The main and most frequently used methods are *finite difference method* (FDM), *method of lines* (MoL) and *finite element method* (FEM) which are described in the following section.

## 3.1 Numerical solution of general partial differential equations

The governing equation (2.26) is the partial differential equation of the second order. These equations are divided into elliptic, hyperbolic and parabolic [43]. Elliptic, hyperbolic or parabolic equations with $n$ variables can be expressed in canonical form as

$$\sum_{k=1}^{n} \frac{\partial^2 u}{\partial x_k^2} + \cdots = 0 \tag{3.1}$$

$$\sum_{k=1}^{n-1} \frac{\partial^2 u}{\partial x_k^2} - \frac{\partial^2 y}{\partial x_n^2} + \cdots = 0 \tag{3.2}$$

$$\sum_{k=1}^{n} \frac{\partial^2 u}{\partial x_k^2} + a_1 \frac{\partial u}{\partial x_1} + \cdots = 0. \tag{3.3}$$

The number of solutions for certain PDE is infinite and thus the general solution concept is established for PDE. To limit the number of solutions, the boundary, in case of non-stationary problems also initial, conditions have to be stated. Equation (2.14) can be taken

as an example where two boundary conditions for space variable ($C(t, \pm\infty) = 0$) and one initial time boundary ($C(0, x) = M/(A\delta x)$) have to be stated. Here, $M$ (kg) is weight of substance, $A$ ($m^2$) is source area and $\delta x$ (m) is Dirac function. In this case, the equation (2.14) has the solution [52]

$$C(x,t) = \frac{M}{\sqrt{4\pi D t}} \exp\left(-\frac{x^2}{4Dt}\right). \tag{3.4}$$

When new partial derivation is added to the equation (e.g. new space dimension), the two appropriate boundary conditions have to be defined.

The details of widely used FDM, MoL and FEM methods are described in the following subsections. Nevertheless, other methods also have been developed for solving PDE, namely boundary element method [11], method of characteristics or coupled Eulerian-Lagrangian method. The latter will be described in more detail in section 3.3.4.

### 3.1.1 Finite difference method

Finite difference method [25] is one of the basic methods for solving partial differential equations. Its idea is to approximate the individual governing terms. The partial derivation terms are substituted by finite differences. For a proper approximation, the space and time is needed to be discretized into the structured matrix of points. The finer matrix the more accurate approximation.

To approximate the derivatives, Taylor series expansion is used. Its basic principle is the following. For $\frac{\partial u}{\partial x}$ derivation, it holds that

$$\frac{\partial u}{\partial x} = \lim_{\Delta x \to 0} \frac{u(x + \Delta x) - u(x)}{\Delta x}. \tag{3.5}$$

The substitution of function $u(x + \Delta x)$ by Taylor series in neighbourhood of point $u(x)$ is defined as

$$u(x + \Delta x) = u(x) + \Delta x \frac{\partial u(x)}{\partial x} + \frac{(\Delta x)^2}{2!} \frac{\partial^2 u(x)}{\partial x^2} + \frac{(\Delta x)^3}{3!} \frac{\partial^3 u(x)}{\partial x^3} + \cdots. \tag{3.6}$$

Rearranging of terms in equation (3.6) and division by $\Delta x$ leads to the following form

$$\frac{u(x + \Delta x) - u(x)}{\Delta x} = \frac{\partial u(x)}{\partial x} + \frac{\Delta x}{2!} \frac{\partial^2 u(x)}{\partial x^2} + \frac{(\Delta x)^2}{3!} \frac{\partial^3 u(x)}{\partial x^3} + \cdots = \frac{\partial u(x)}{\partial x} + \mathcal{O}(\Delta x), \tag{3.7}$$

where $\mathcal{O}(\Delta x)$ is truncation error.

Equation (3.7) expresses the approximation of derivation $\frac{\partial u}{\partial x}$ where the error being dropped is proportional to size of $\Delta x$. Equation (3.7) is labelled as forward difference due to the usage of neighbour terms. Backward and central differences of the first derivative can be defined in similar way as

$$\frac{\partial u(x)}{\partial x} = \frac{u(x) - u(x - \Delta x)}{\Delta x} + \mathcal{O}(\Delta x), \tag{3.8}$$

$$\frac{\partial u(x)}{\partial x} = \frac{u(x + \Delta x) - u(x - \Delta x)}{2\Delta x} + \mathcal{O}((\Delta x)^2). \tag{3.9}$$

Similarly, the derivations of the second-order can be approximated by finite difference formulas. The most commonly used is the central difference in the form

$$\frac{\partial^2 u}{\partial x^2} = \frac{u_{i+1} - 2u_i + u_{i-1}}{\Delta x^2} + \mathcal{O}((\Delta x)^2). \tag{3.10}$$

The important properties that influence the approximation accuracy of FDM are especially consistence, stability and convergence [11]. The approximating equation is consistent when it becomes an originally approximated PDE in case the step sizes are approaching zero. Consistency is generally satisfied when the approximations are based on Taylor expansion.

The numerical scheme of FDM is stable when the error lies in finite boundaries. Finally, convergence of FDM is satisfied when its solution approaches the solution of approximated PDE in case step sizes are approaching zero. Both consistency and stability are needed preconditions for convergence.

### 3.1.2 Method of lines

Basic idea of method of lines (MoL) [26] is similar to FDM - to approximate partial derivatives (space derivatives in most cases) by algebraic expressions and to leave one derivation (time in most cases) untouched. After this procedure, partial differential equation is transformed to a system of ordinary differential equations (ODE). The most common way how to approximate partial derivations is to use finite differences as described in section 3.1.1.

ODE can be solved by variety of methods that have been developed over the last century. In order to describe the most commonly used methods, the initial value problem is taken into account

$$u'(t) = f(t, u(t)), \quad u(t_0) = u_0, \tag{3.11}$$

where $u(t)$ is function to be found, $u'(t)$ is its time derivation and $u(t_0) = u_0$ is initial condition in time $t_0$.

When solving initial value problem, it is important that there does exist exactly and only one solution. It is well known that the condition of existence and uniqueness of solution is satisfied when function $f(t, u(t))$ is continuous, bounded and satisfies Lipschitz condition in all its points.

**Definition 3.1.1.** *Function $f$ satisfies Lipschitz condition in point $(t_0, u_0)$ if there exist constant $L$ and boundary $B$ of point $(t_0, u_0)$ such as*

$$|f(t, a) - f(t, b)| \leq L\,|a - b|\,, \quad \forall (t, a), (t, b) \in B. \tag{3.12}$$

The numerical solution of ODE is the calculation of approximation values of function $u(t)$ in certain points $t_i$ which lie in certain interval $[a, b]$. Let the calculated approximation of exact solution $u(t_i)$ in point $t_i$ be labelled as $w_i$. The numerical methods for solving ODE can be split into two groups according to the way of calculation:

1. single step methods use only the values of $w_i$ for $w_{i+1}$ approximation for $i = 0, 1, ...$
2. multi-step methods ($k$-step methods) use the previously calculated values $w_i, w_{i-1}$, $..., w_{i-k+1}$ for $w_{i+1}$ approximation.

The calculation is done iteratively. In case of multi-step methods, the initial calculation could be difficult because the values from preceding step(s) are not known yet. For its initialization, a single step method has to be used. In the following subsections, the basic and widely used methods to solve ODE are described.

### Taylor series method

Taylor series method is the basic single step method and its basic form is

$$w_{i+1} = w_i + hw_i' + \frac{h^2}{2!}w_i'' + \cdots + \frac{h^n}{n!}w_i^{(n)} \tag{3.13}$$

where $h$ is integration step. The method is the basic method for all single step methods. It provides the most precise function approximation. The number of terms is not limited in general. In practical usages, the two last calculated terms can be directly used to define truncation error of the calculation, therefore the criteria to finish the current iteration step is very clear. On the other hand, the biggest problem of the method is that higher derivatives of function have to be calculated [2].

### Explicit Euler method

Explicit Euler method is the simplest single step method directly derived from Taylor series (3.13). The form of the method is

$$w_{i+1} = w_i + hf(t_i, w_i), \tag{3.14}$$

where $h$ is integration step.

Explicit Euler method uses first two terms of Taylor series method and it belongs to first-order methods. The method is less precise which can be compensated by decreasing the step size, however, when the step is very low truncation errors will grow and the overall error will become larger.

### Midpoint method

Midpoint method extends explicit Euler method with intermediate calculation step in the middle of integration step

$$w_{i+1} = w_i + hf\left(t + \frac{h}{2}, w_i + \frac{h}{2}f(t, w_i)\right), \tag{3.15}$$

where $h$ is integration step. Midpoint method is the simplest one from family of Runge-Kutta methods.

### Runge-Kutta methods

Runge-Kutta methods are a generalization of midpoint method with arbitrary number of inner points/steps. Its general form can be expressed as

$$w_{i+1} = w_i + h\sum_{i=1}^{s} b_i k_i, \tag{3.16}$$

where

$$\begin{aligned}
k_1 &= f(t_i, w_i), \\
k_2 &= f(t_i + c_2 h, w_i + a_{21} h k_1), \\
k_3 &= f(t_i + c_3 h, w_i + a_{31} h k_1 + a_{32} h k_2), \\
&\vdots \\
k_s &= f(t_i + c_s h, w_i + a_{s1} h k_1 + a_{s2} h k_2) + \ldots + a_{s,s-1} h k_{s-1}.
\end{aligned}$$

The well known and mostly used Runge-Kutta method is of the fourth order in the following form

$$w_{i+1} = w_i + \frac{1}{6}h(k_1 + 2k_2 + 2k_3, +k_4), \tag{3.17}$$

$$\begin{aligned} k_1 &= f(t_i, w_i), \\ k_2 &= f(t_i + \frac{1}{2}h, w_i + \frac{1}{2}hk_1), \\ k_3 &= f(t_i + \frac{1}{2}h, w_i + \frac{1}{2}hk_2), \\ k_4 &= f(t_i + h, w_i + hk_3). \end{aligned}$$

The method described by equation (3.17) is very popular because of its relative simplicity, good stability and precision. Because of this, Runge-Kutta methods are still mostly used. Their comprehensive overview is written by Butcher [8].

### Adams-Bashforth method

Adams-Bashforth method belongs to the family of explicit multi-step methods and its form is as follows

$$w_{i+1} = w_i + h\sum_{j=1}^{k} b_j f(t_{i+1-j}, w_{i+1-j}), \tag{3.18}$$

where $h$ is integration step and $b_j$ are given coefficients. Using only the first coefficient ($k = 1$) leads to the previously defined explicit Euler method.

### Implicit Euler method

Implicit methods are used in cases when explicit methods are not stable and thus their calculation is not precise or it is divergent. Implicit Euler method is the simplest implicit method and is defined as

$$w_{i+1} = w_i + hf(t_{i+1}, w_{i+1}), \tag{3.19}$$

where $h$ is integration step.

In case of ODE system of the first order, the method leads to solving the system of algebraic equations.

### Crank-Nicolson method

Crank-Nicolson method combines implicit and explicit Euler methods. The method can be also derived from implicit Runge-Kutta methods of the second order [47]. The equation has the following form

$$w_{i+1} = w_i + h\frac{1}{2}(f(t_i, w_i) + f(t_{i+1}, w_{i+1})), \tag{3.20}$$

where $h$ is integration step.

### 3.1.3 Finite element method

Finite element method (FEM) [11] searches, in opposite to FDM, the approximation of solution $u$ of the original PDE. The idea of FEM is to discretize the space into smaller parts, so called elements, and to find the functions that are the solutions on particular elements. The sum of all these functions whose domain equals to the single elements is the searched solution. The values of the functions have to be the same on the common points (nodes) of each two elements.

The elements are defined as lines in one-dimensional case. The example of such element is given in figure 3.1. The linear *trial functions* $\Phi_n^{(e)}(x)$, where $n$ is the index of element $e$, are used there. The trial functions interpolate function value $u^{(e)}$ between element nodes. The neighbouring elements with function $u^{(e)}$ and with common node are shown in figure 3.2.

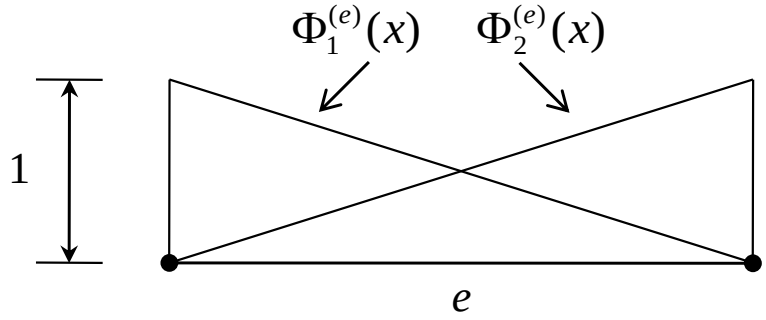



Figure 3.1: Element with two trial functions $\Phi_n^{(e)}(x)$.

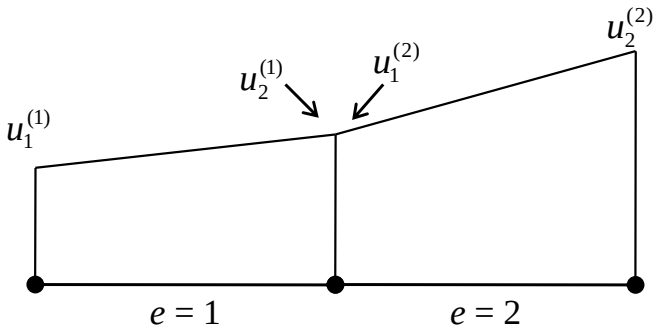



Figure 3.2: Two elements with two functions $u^{(e)}$ and with one common node.

Function $u^{(e)}$ defined on the element $e$ is expressed as $u^{(e)}(x) = \Phi_n^{(e)}(x)u_n^{(e)}$. Overall function $u$ in whole domain has the form

$$u = \bigcup_{e=1}^{E} \Phi_n^{(e)}(x)u_n^{(e)}, \tag{3.21}$$

where $E$ is the number of elements in the domain.

What remains is to find functions $u^{(e)}$. Weighted residual method where the residual (error) of differential equation $R$ is minimized is used for this purpose. So called *testing*

*functions* $g$, which define subspace the residual $R$ is projected on, are introduced. This process is known as scalar product of testing function and residual and can be expressed as

$$(g, R) = \int_0^1 gR dx = 0, \quad 0 < x < 1. \tag{3.22}$$

The substitution of function $u$ for $R$ into equation (3.22) and integration by parts lead to so called *weak form* of governing equations. The result of the process is the system of algebraic equations. This system has a tridiagonal form in one-dimensional case.

If the constant function is substituted for $g$ in equation (3.22) then the method is called finite volume method (FVM). This leads to simplification, therefore this method is often used [4].

The selection of various trial and testing functions defines different types of FEM methods. The example can be Galerkin method which uses piecewise linear trial and the same testing functions. The functions are not only linear or piecwise linear but also quadratic or polynomial.

The advantage of FEM over FDM is the ability to solve the equations in space with 'general' geometry and natural incorporation of boundary conditions of Neumann type (see below) [31].

## 3.2 Terms and properties of ADE equations

Previously, the general frameworks for solving partial differential equations were described. In this section the attention is given to basic terms and properties of ADE equations which are special case of general PDE.

### 3.2.1 Boundary conditions

In practice, ADE and other PDE equations are computed in region $\Omega$ that is finite and on its boundary, denoted as $\Gamma = \partial\Omega$, it is assumed that function $u$ satisfies appropriate boundary conditions. A partial differential equation with the given initial and boundary conditions is called an *initial-boundary value problem.*

Lets consider the partitioning of boundary $\Gamma$ into $\Gamma_D, \Gamma_N$ and let $\gamma_D, \gamma_N$ be given functions on the corresponding boundary parts. The typical boundary conditions are *Dirichlet condition*

$$u = \gamma_D \quad \text{on } \Gamma_D, \tag{3.23}$$

and *Neumann condition*

$$\frac{\partial u}{\partial \vec{n}} = \gamma_N \quad \text{on } \Gamma_N, \tag{3.24}$$

where $\vec{n}$ is unit normal vector on boundary $\Gamma_N$ pointing outward from region $\Gamma$. Occasionally, the *mixed condition*, also called *Robin (Newton) condition*, can be stated [26]. The description how the boundary conditions are resolved on structured grids for pure advection and diffusion problems follows. There, $u_a$ and $u_{aa}$, which are used for simplicity, have the meanings of $\frac{\partial u}{\partial a}$ and $\frac{\partial^2 u}{\partial a^2}$.

### Advection equation

Dealing with boundary conditions in practise leads to use virtual points outside the grid. Lets consider one-dimensional advection equation $u_t + u_x = 0$ for $t > 0$ and grid interval $x \in (0, 1)$ with given initial condition $u(x, 0)$ and inflow boundary condition $u(0, t) = \gamma_0(t)$. Lets further consider the spatial discretization using central difference on the grid $\Omega_h = x_j$ with nodes $x_j = jh, j = 1, ..., m$ and mesh width $h = 1/m$. The semi-discrete system will have the form

$$w_j'(t) = \frac{1}{2h}\left(w_{j-1}(t) - w_{j+1}(t)\right), \quad j = 1, 2, ..., m, \tag{3.25}$$

where $w_0(t) = \gamma_0(t)$ and $w_{m+1}(t)$ represents approximation at the *virtual point* $x_{m+1} = jm + h = 1 + h$. The value can be found by extrapolation using the internal grid points near the boundary. A case of linear extrapolation using two points leads to

$$w_{m+1}(t) = 2w_m(t) - w_{m-1}(t). \tag{3.26}$$

### Diffusion equation

Lets now consider the diffusion equation in the form $u_t - u_{xx} = 0$ for $t > 0$ and grid interval $x \in (0, 1)$ with given initial condition $u(x, 0)$, Dirichlet boundary condition $u(0, t) = \gamma_0(t)$ for the left point and Neumann boundary condition $u_x(1, t) = 0$ for the right point. Lets further assume the uniform grid $\Omega_h$ with nodes $x_j = jh, j = 1, ..., m$ and mesh width $h = 1/m$ is used. The second-order central difference semi-discrete scheme has the form

$$w_j'(t) = \frac{1}{h^2}\left(w_{j-1}(t) - 2w_j(t) + w_{j+1}(t)\right), \quad j = 1, 2, ..., m, \tag{3.27}$$

where $w_0(t) = \gamma_0(t)$ and $w_{m+1}(t)$ represents approximation at the virtual point $x_{m+1} = jm + h = 1 + h$ and it is to be determined by the following Neumann condition

$$\frac{1}{2h}\left(w_{m+1}(t) - w_{m-1}(t)\right) = 0. \tag{3.28}$$

Neumann condition as defined in the example is homogeneous. If the central difference formula (3.28) is used to approximate the value $w_{m+1}(t)$ it leads to a simple equation

$$w_{m+1}(t) = w_{m-1}(t). \tag{3.29}$$

Equation (3.29) presents a natural choice because of the symmetry nature of the homogeneous Neumann condition $u_x(1, t)$. If the grid is extended to $x \in (0, 2)$ and let function $\tilde{u}$ on this grid is defined as

$$\tilde{u}(x, t) = \begin{cases} u(x, t)) & \text{for } 0 \leq x \leq 1 \\ u(1 - x, t) & \text{for } 1 < x \leq 2 \end{cases}, \tag{3.30}$$

then the problem for $\tilde{u}$ is defined as $\tilde{u}_t - \tilde{u}_{xx} = 0$ with $\tilde{u}(0, t) = \tilde{u}(2, t) = \gamma_0(t)$ and the homogeneous Neumann condition from the previous problem $\tilde{u}(1, t) = 0$ is automatically fulfilled due to the symmetry around point $x = 1$ and $\tilde{w}_{m+j}(t) = \tilde{w}_{m-j}(t)$ will automatically hold after discretization.

### Cell/vertex centered grids

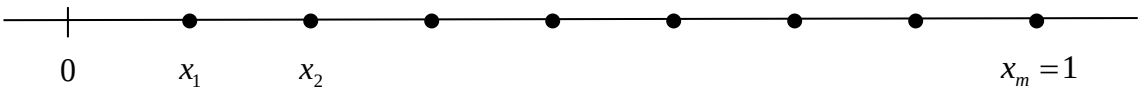
Lets consider the example of the second-order semi-discrete system reprezented by equation (3.27) and the central difference approximations of the spatial derivative $u_{xx}$ (3.28). Further, lets consider the definition of boundary conditions in the form

$$u(0,t) = \gamma_0(t), \tag{3.31}$$

$$u_x(1,t) = \gamma_1(t) \tag{3.32}$$

for $t > 0$, $0 < x < 1$ and with initial profile $u(x,0)$.

The *vertex centered grid* with nodes $x_j = jh$ and $h = 1/m$ is defined in the same way like in the previous case of finite difference grid (see below).




To formulate semi-discrete system which solves the diffusion equation the right hand side of Dirichlet boundary condition (3.31) is substituted into function value of boundary point 0 ($w_0 = \gamma_0(t)$) and $w_{m+1}$ function value is determined using the discretization (3.28) of Neumann boundary condition (3.32). Then, the following semi-discrete system is obtained on vertex centered grid

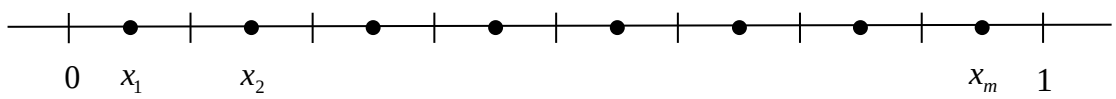
$$\begin{aligned} w_1'(t) &= \frac{1}{h^2}(-2w_1(t) + w_2(t)) + \frac{1}{h^2}\gamma_0(t), \\ w_j'(t) &= \frac{1}{h^2}(w_{j-1}(t) - 2w_j(t) + w_{j+1}(t)), \quad j = 2, ..., m-1, \\ w_m'(t) &= \frac{1}{h^2}(2w_{m-1}(t) - 2w_m(t)) + \frac{2}{h}\gamma_1(t). \end{aligned} \tag{3.33}$$

It can be found out [26] that truncation error $\sigma_{h,j}(t)$ of the above system is $\mathcal{O}(h^2)$ in all nodes except at right boundary node $x_m$ where it is

$$\sigma_{h,m}(t) = \frac{1}{3}h\frac{\partial^3 u}{\partial x^3}(1,t) + \mathcal{O}(h^2).$$

In case of homogeneous Neumann condition $\frac{\partial^3 u}{\partial x^3}(1,t) = 0$ holds and the error $\sigma_{h,m}(t)$ drops from $\mathcal{O}(h)$ to $\mathcal{O}(h^2)$ which is not the case of non-homogeneous condition.

Lets now consider the *cell centered grid* with nodes $x_j = (j - \frac{1}{2})h$, $h = 1/m$ (see below).




Now the right boundary condition fits in natural way if the difference formula is interpreted as flux form. The difference quotients for right boundary are $\frac{1}{h}(w_{m+1} - w_m) = \gamma_1$. However, the left Dirichlet boundary condition definiton is not straightforward. The virtual

value $w_0$ at virtual point $x_0 = -\frac{1}{2}h$ is extrapolated from boundary condition value $w_1$ and boundary value $\gamma_0$ and thus can be defined as $w_0 = 2\gamma_0 - w_1$. Putting all together leads to the following semi-discrete system

$$\begin{aligned} w_1'(t) &= \frac{1}{h^2}(-3w_1(t) + w_2(t)) + \frac{2}{h^2}\gamma_0(t), \\ w_j'(t) &= \frac{1}{h^2}(w_{j-1}(t) - 2w_j(t) + w_{j+1}(t)), \quad j = 2, ..., m-1, \\ w_m'(t) &= \frac{1}{h^2}(w_{m-1}(t) - w_m(t)) + \frac{1}{h}\gamma_1(t). \end{aligned} \tag{3.34}$$

The truncation errors on all nodes except the boundaries are again of $\mathcal{O}(h^2)$. The truncation error at the left boundary node is

$$\sigma_{h,1}(t) = \frac{1}{4}hu_{xx}(0,t) + \mathcal{O}(h^2),$$

at the right boundary it is

$$\sigma_{h,m}(t) = \frac{1}{24}h\frac{\partial^3 u}{\partial x^3}(1,t) + \mathcal{O}(h^2).$$

Obviously, $\sigma_{h,m}$ is lower for cell centered grid and $\sigma_{h,1}$ is lower for vertex centered grid. If the problem being solved leads to this case the cell/vertex discretizations can be combined taking $h = 1/(m + \frac{1}{2})$ and $x = jh$ and form the *hybrid grid*.

### 3.2.2 Monotonicity

Advection-diffusion equation underlies the laws of mass conservation, therefore if $u(x,t)$ is interpreted as a concentration of some species then the integral $M(t)$ defined by equation (3.35) represents the mass on interval $[a,b]$. It can be shown that $\frac{d}{dt}M(t) = 0$ and thus mass balance is preserved [26].

$$M(t) = \int_a^b u(x,t)dx. \tag{3.35}$$

#### Positivity

Because of the physical nature of ADE equations whose solutions are concentrations of chemical species, it is natural that the following holds

$$u(x,0) \geq 0 \quad \text{for all } x \quad \Rightarrow \quad u(x,t) \geq 0 \quad \text{for all } x \text{ and } t > 0. \tag{3.36}$$

In general, there is no guarantee that the spatial discretization maintains the above non-negative property. Thus it is desirable if the discrete or semi-discrete system can be evaluated with this respect. Lets consider the system of ordinary differential equations in $\mathbb{R}^m$ for $t > 0$

$$w'(t) = F(t, w(t)). \tag{3.37}$$

This system will be called *positive* if

$$w(0) \geq 0 \quad \Rightarrow \quad w(t) \geq 0 \quad \text{for all } t > 0. \tag{3.38}$$

To define the criteria telling the system (3.37) is positive lets consider it is written in a matrix form

$$w'(t) = Aw(t), \tag{3.39}$$

where $A$ is real $m \times m$ *circulant* matrix defined as

$$A = \begin{bmatrix} a_0 & a_1 & . & . & a_{m-1} \\ a_{m-1} & a_0 & a_1 & . & a_{m-2} \\ . & a_{m-1} & a_0 & . & . \\ . & . & . & . & a_1 \\ a_1 & a_2 & . & a_{m-1} & a_0 \end{bmatrix}. \tag{3.40}$$

As it is shown in section 3.3, circulant matrices arise in discretizations of PDE (and thus also ADE) with periodical boundary conditions. To determine if ODE system is positive, the following theorem can be used.

**Theorem 3.2.1.** *The linear system $w'(t) = Aw(t)$ is positive if and only if*

$$a_{ij} \geq 0 \quad \textit{for all } i \neq j. \tag{3.41}$$

The proof can be found in [26].

Theorem 3.2.1 can be directly used to decide whether given discretization scheme is positive or not. It will be further shown that first-order upwind scheme for pure advection problem is positive, however, higher order upwind schemes are not since they contain negative coefficients outside the diagonal line in matrix $A$. Similarly, the second-order central difference scheme is positive for pure diffusion equation and the fourth-order is not due to the same reason.

#### Maximum principle

Another very important property of ADE equations is called the *maximum principle* which can be expressed, assuming $N$ is number of grid points, as

$$\min_j w_j(0) \leq w_i(t) \leq \max_j w_j(0) \quad \forall t \geq 0, \forall i, j \in \{1, 2, ..., N\}. \tag{3.42}$$

The maximum principle says that the concentration anywhere in space $[a, b]$ cannot be lower/larger than minimum/maximum concentration value of the initial concentration profile $u(x, 0)$. The maximum principle property means that there are no *global overshoots* or *undershoots* in the system. Again, the experiments in section 3.3 show that this principle is also not always satisfied.

### 3.2.3 Flux and its limiting

As it can be seen in the previous section and in experiment results in section 3.3 some of the numerical schemes (mainly for advection equation part) can cause oscillations around the exact solution which can lead also to global overshoots or undershoots. The problem is seen mostly in higher-order schemes like third-order upwind biased scheme. The reason for such behaviour is inaccurate calculation of local fluxes between grid points or cells. Therefore, the description of the flux form of the ADE equation and its limitation is further discussed.

### Flux form of equation

To find out the cause of such behavior lets consider the discretization of pure advection equation (2.15). The space is discretized into the uniform vertex centered grid $\Omega_h$. Lets further consider the auxiliary grid points $x_{j\pm1/2} = \frac{1}{2}(x_{j\pm1} + x_j)$ lying in the middle of grid points. The auxiliary grid points $x_{j\pm1/2}$ now delimit the grid cell $\Omega_j = \left[x_{j-1/2}, x_{j+1/2}\right]$. Further, the cell averages are defined as

$$\overline{u}(x_j, t) = \frac{1}{h}\int_{\Omega_j} u(x,t)dx = u(x_j,t) + \mathcal{O}(h^2).$$

The moving mass into/out from/to cell is called *inflow/outflow flux*. It is physically correct to assume that the cell averages can change only by moving the concentrations out or into the cell. At each cell interface $x_{j\pm1/2}$, fluxes $f_{j\pm1/2}$ can be formulated and advection equation can be written as

$$h\frac{d}{dt}\overline{u}(x_j,t) = f_{j-\frac{1}{2}}\left(t, u_{j-\frac{1}{2}}(t)\right) - f_{j+\frac{1}{2}}\left(t, u_{j+\frac{1}{2}}(t)\right), \tag{3.43}$$

where $f_{j\pm1/2}(t, u_{j\pm1/2}(t)) = a(x_{j\pm1/2})u(x_{j\pm1/2}, t)$ and $a$ is advection coefficient. The equation (3.43) is the *flux form* of advection equation (2.15).

It is quite straightforward to define flux form for semi-discrete system

$$w_j'(t) = \frac{1}{h}\left(a(x_{j-\frac{1}{2}})w_{j-\frac{1}{2}}(t) - a(x_{j+\frac{1}{2}})w_{j+\frac{1}{2}}(t)\right), \quad j = 1, ..., m, \tag{3.44}$$

where $w_{j\pm1/2}(t)$ are approximated values in cell boundaries. The choice of $w_{j\pm1/2}(t)$ in terms of neighbouring nodes determines the actual discretization.

### Flux limiting

Higher-order schemes can be viewed as first-order schemes, characterized by low accuracy, with some correction function. The corrections can be often simply too large and can result in oscillations around the exact solution. To make the further discussion more evident lets assume the case of the third-order biased scheme (3.75, 3.76). The approximation value $w_{j+1/2}(t)$ on the cell boundary can be determined by interpolation and has the form

$$w_{j+\frac{1}{2}}(t) = \begin{cases} \frac{1}{6}(-w_{j-1}(t) + 5w_j(t) + 2w_{j+1}(t)), & \text{if } a(x_{j+\frac{1}{2}}) \geq 0, \\ \frac{1}{6}(2w_j(t) + 5w_{j+1}(t) - w_{j+2}(t)), & \text{if } a(x_{j+\frac{1}{2}}) < 0, \end{cases} \tag{3.45}$$

with the fluxes $a(x_{j\pm1/2})w_{1+1/2}(t)$. For case of $a > 0$ the flux will have the form

$$f_{j+\frac{1}{2}}(t, w) = \frac{1}{6}a(-w_{j-1} + 5w_j + 2w_{j+1}). \tag{3.46}$$

By defining $\theta_j$ as ratio

$$\theta_j = \frac{w_j - w_{j-1}}{w_{j+1} - w_j},$$

equation (3.46) can be rewritten in the following form

$$f_{j+\frac{1}{2}}(t, w) = a\left(w_j + \left(\frac{1}{3} + \frac{1}{6}\theta_j\right)(w_{j+1} - w_j)\right). \tag{3.47}$$

Equation (3.47) can be generalized into

$$f_{j+\frac{1}{2}}(t,w) = a(w_j + \psi(\theta_j)(w_{j+1} - w_j)), \quad a > 0, \tag{3.48}$$

with $\psi$ entitled as *limiter function*. The limiter function is chosen such as it has better accuracy than the first-order upwind scheme but still preserves the positivity property. It can be deduced [26] that to maintain positivity it is sufficient for limiter function $\psi$ presented in equation (3.48) that

$$0 \leq \psi(0) \leq 1, \quad 0 \leq \frac{1}{\theta}\psi(\theta) \leq \mu \quad \text{for all } \theta \in \mathcal{R}, \tag{3.49}$$

where $\mu$ is any positive real number, however, the evaluations revealed that 1 is a reasonable value.

An example of piece-wise linear limiter function is the one introduced in [29]

$$\psi(\theta) = \max\left(0, \min\left(1, \frac{1}{3} + \frac{1}{6}\theta, \theta\right)\right). \tag{3.50}$$

To illustrate the effect of the limiter the simple experiment was done using the third-order upwind biased scheme with the same conditions as the series of experiments in section 3.3. The results after applying the limiter can be seen in figure 3.3 where horizontal axis $x$ is one-dimensional position and vertical axis $c$ states function values (concentrations) at position $x$ and final time $t = 1$. It is obvious that limiting has its negative aspect of adding small amount of diffusivity to the scheme. This type of flux limiting can be seen as a trade-off between accuracy and positivity.

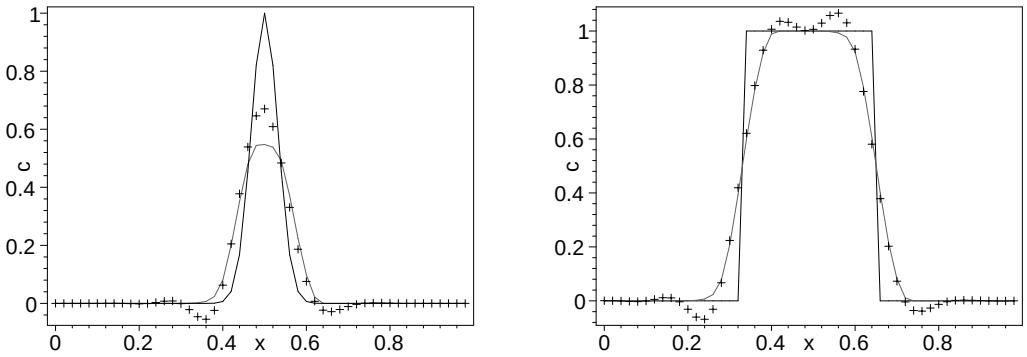



Figure 3.3: Test result of the third order upwind-biased scheme for sine- (left) and box-shape (right) initial profiles. Solid lines show the result of the limited and point lines with crosses show the results of the original scheme.

The flux limiting in a sense described here is not the only approach. When more specialized schemes are used the actual form of the limiting approach is adjusted as it can be seen in case of Walcek method described in section 3.3.3.

### 3.2.4 Non-uniform and multi-dimensional grids

Although, non-uniform grids are out of the scope of this thesis, their concept is presented here due to their necessity for practical and complex problems.

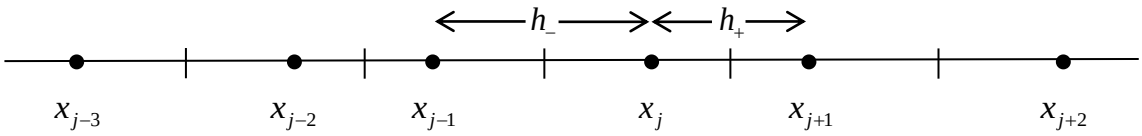
The definition of a non-uniform grid can be easily shown for one-dimensional case. Lets consider the vertex centered grid where the cell vertices $x_{j+1/2}$ are centered between the grid points $x_j$ and $x_{j+1}$

$$x_{j+\frac{1}{2}} = \frac{1}{2}(x_j + x_{j+1}), \quad h_j = \frac{1}{2}(x_{j+1} - x_{j-1}).$$

For convenience of notation let $h_\pm$ be defined as

$$h_- = x_j - x_{j-1}, \quad h_+ = x_{j+1} - x_j,$$

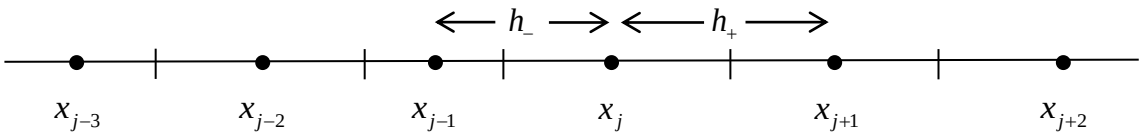
then the vertex centered non-uniform grid in one dimension can be visualized as below.




Similarly, the uniform cell centered grid can be generalized into its non-uniform form. The one-dimensional spatial domain is discretized into cells $\Omega_j = [x_{j-1/2}, x_{j+1/2}]$ such as the grid points $x_j$ are cell centers. It follows that

$$x_j = \frac{1}{2}(x_{j-\frac{1}{2}} + x_{j+\frac{1}{2}}), \quad h_j = x_{j+\frac{1}{2}} - x_{j-\frac{1}{2}}$$

and $h_\pm$ are defined as before. The visualization of such grid is shown below.




The schemes defined for non-uniform grids either for vertex (finite differences) or cell (finite volumes) centered grids show the similar stability and accuracy. The main advantage of cell centered grids in general is the treatment with a tracer and its concentration in physical sense and thus the possibility to limit the fluxes in cell boundaries as shown in section 3.2.3. Moreover, the higher order upwind schemes can be derived in quite straightforward manner [26].

Arbitrary non-uniform grids bring additional difficulties to the numerical schemes. They negatively influence stability and even more the consistency of the solution and thus the overall convergence of the schemes. Non-uniformity can be restricted in a smoothness sense and hence *smooth grid* can be defined. The non-uniform grid is smooth when the local differences of the mutual spatial steps $h_-$ and $h_+$ cause errors only of the second order.

### Finite elements

In section 3.1.3, the general form of finite element framework was introduced. Due to the local nature of the FEM discretization when the local trial functions (3.21) limits the solution to local elements, FEM discretization can be naturally extended to non-uniform grids.

Galerkin method, a basic representative of FEM, uses linear test functions in its definition (see figure 3.1). The method produces quite big oscillations around the exact solution. There exist numerous modifications of this basic FEM version such as Petrov-Galerkin methods which introduce the tilting of linear test functions in upwind directions which leads to almost non-oscillating solutions mainly for stationary problems [26].

#### Multi-dimensional Cartesian grids

The uniform one-dimensional grids can be easily extended to its multi-dimensional form, called *Cartesian grid* [26]. ADE equation written in multiple spatial dimensions $d$ using Cartesian coordinates $\vec{x} = (x_1, x_2, ..., x_d)^T \in \mathcal{R}^d$ reads

$$u_t + \sum_{k=1}^{d} a_k u_{(x_k)} = \sum_{k=1}^{d} D_k u_{(x_k)(x_k)} + S(\vec{x}, t),$$

where $S(\vec{x}, t)$ (kg m$^{-d}$ s$^{-1}$) is source term. The rest of the thesis will deal at most with the two-dimensional problems ($d = 2$) which are sufficient to show the performance of presented schemes and yet easier to implement than the corresponding three-dimensional versions.

The extension of finite difference methods into multiple dimensions using Cartesian grids is straightforward. Each dimension is treated separately and the unknowns in two dimensions are seen as point values in grid points $(x_i, y_j)$.

The schemes derived within the finite volume framework can be also derived for two dimensions. Here, the unknowns are viewed as averages over cells $\Omega_{ij}$ with size $[\Delta x \times \Delta y]$ and defined in uniform grids as

$$\Omega_{ij} = \left[x_i - \frac{1}{2}\Delta x, x_i + \frac{1}{2}\Delta x\right] \times \left[y_j - \frac{1}{2}\Delta y, y_j + \frac{1}{2}\Delta y\right].$$

Lets consider two-dimensional advection-diffusion problem

$$u_t + f(u, u_x)_x + g(u, u_y)_y = 0$$

in conservation form and with advection-diffusion fluxes $f$ and $g$ in $x$ and $y$ dimensions. Then, the flux form for semi-discrete system in two dimensions will have the similar form as equation (3.44)

$$w'_{ij}(t) = \frac{1}{\Delta x}\left(f_{i-\frac{1}{2},j}(t) - f_{i+\frac{1}{2},j}(t)\right) + \frac{1}{\Delta y}\left(g_{i,j-\frac{1}{2}}(t) - g_{i,j+\frac{1}{2}}(t)\right).$$

Many numerical methods were designed to deal with unstructured grids (e.g. triangles in two dimensions) for practical calculations, especially within FEM framework. Because the topic of unstructured grids is out of the scope of this thesis the reader is referred to specialized literature, for instance [4], [14] or [11].

### 3.2.5 Operator splitting for ADE equations

Very often, ADE equations have complex form and the so called *operator splitting* technique is used to decompose ADE to its advection and diffusion parts and to solve these parts independently with specialized schemes. Moreover, advection equations are solved often by explicit schemes where the time step is limited because of stability and its size has to

pass Courant–Friedrichs–Lewy (CFL) [30] condition which is more strict when using more dimensions. CFL number is defined as

$$CFL = \sum_{i=1}^{d} \frac{a_{max}\Delta t}{\Delta h_d} \leq CFL_{max}, \tag{3.51}$$

where $d$ is number of dimensions, $a_{max}$ (m s$^{-1}$) is maximum wind speed in domain, $\Delta t$ is time (scheme) step (s), $h_d$ (m) is step size in dimension $d$ and $CFL_{max}$ is threshold which is specific for each numerical scheme (typically 1). The less dimensions are involved in actual scheme the less strict CFL condition is. Therefore, the dimensional splitting, if possible, is often advocated.

Lets consider advection-diffusion equation (3.52) which is split to advection (3.53) and diffusion (3.54) parts where the results from advection solution can serve as an input to calculate the diffusion solution and vice versa.

$$u_t + au_x = Du_x x, \tag{3.52}$$

$$u_t = -au_x, \tag{3.53}$$

$$u_t = Du_x x, \tag{3.54}$$

However, the operator splitting may also involve the *splitting error*. This error can be partly reduced when using the *Strang splitting* technique which description follows. Let $S(\Delta t)$ denotes the solution operator to ADE equation (3.52) during one time step of size $\Delta t$, thus the scheme to solve ADE will be of the form

$$w(t + \Delta t) = S(\Delta t)w(t). \tag{3.55}$$

Further, let $S_a(\Delta t)$ and $S_d(\Delta t)$ be the solution operators to the advection (3.53) and diffusion (3.54) equations over time step $\Delta t$. Then, Strang splitting for the two operators is defined as

$$w(t + \Delta t) = S_a(\Delta t/2)S_d(\Delta t)S_a(\Delta t/2)w(t). \tag{3.56}$$

The operators $S_a$ and $S_d$ can be theoretically exchanged, however, in practice it is better to give smaller time step to advection scheme. It has been shown that in general symmetric Strang splitting technique results in the splitting error of $\mathcal{O}((\Delta t)^2)$ which is of course better than $\mathcal{O}(\Delta t)$ in case of non-symmetric splitting [53]. Therefore, Strang splitting is advocated in practical calculations.

There exist certain scenarios when splitting error is zero. It is in general the case when the two splitting operators commute [53]. Lanser & Vewer [30] formulated the scenario when this holds for advection-diffusion splitting: *Advection commutes with diffusion if* $a$ *and* $D$ *are independent on spatial dimensions.* In practical computations nor velocity field neither diffusion matrix are independent on dimension variables, however, the extended use of Strang splitting in computational air pollution modeling leads to the conclusion that in this field splitting errors are kept within reasonable bounds.

### 3.2.6 Error measurement and evaluation

The experiments serving to evaluate the models or the schemes, which are actually used to calculate their solution, differ a lot. Depending on the way what data does exist and what is the form of exact solution the approximation is compared to, the different error measures are defined.

The first one relates to law of mass conservation. As stated in equation (3.35), the final mass amount in the experiment should be the same as at the beginning. Taking the discretized domain into account, the mass conservation law for one-dimensional case and interval $[a, b]$ can be written in the form

$$M(t) = \sum_{j=a}^{b} w(x_j, t_0) = \sum_{j=a}^{b} w(x_j, t), \quad \text{for } t > t_0, \tag{3.57}$$

where $x_j$ is value $x$ at the point with index $j$, $w(x_j, t)$ are values at points $(x_j, t)$ calculated by the given scheme. The actual error of mass preservation (MassE) can be expressed as a ratio between mass at $t = t_0$ and mass at any $t > t_0$.

#### $L_n$-norm errors

Mathematically, a norm is a total size or length of all vectors in a vector space or matrices [42], [26]. Let $\vec{x} = (x_1, x_2, ..., x_M)$ is a vector in $M$-dimensional real space $\Omega$ ($x_j \in \mathcal{R}$). Then $l_n$ norm of $\vec{x}$ is defined as

$$\|\vec{x}\| = \sqrt[n]{\sum_{j=1}^{M} |x_j|^n}. \tag{3.58}$$

Several concrete norms are actually used when evaluating the results (vector of values $\vec{r}$) of experiments when exact solution (vector of values $\vec{e}$) is known ($|\vec{r}| = |\vec{e}| = M$). The first one is $l_1$-norm, often called *Manhattan*, when the difference of two vectors is analysed

$$\|\vec{r} - \vec{e}\|_1 = \sum_{j=1}^{M} |r_j - e_j|. \tag{3.59}$$

It is better to normalize error norms to unit vector to overcome the problems of different vector lengths. Then $l_1$-error, called *mean absolute error* (MAE), will have the form

$$MAE(\vec{r}, \vec{e}) = \frac{1}{M}\|\vec{r} - \vec{e}\|_1 = \frac{1}{M}\sum_{j=1}^{M} |r_j - e_j|. \tag{3.60}$$

Similarly to $l_1$-error, $l_2$-error, called *mean squared error* (MSE), can be defined as

$$MSE(\vec{r}, \vec{e}) = \frac{1}{M}\|\vec{r} - \vec{e}\|_2^2 = \frac{1}{M}\sum_{j=1}^{M} (r_j - e_j)^2. \tag{3.61}$$

The last $l_n$-error measure used in the thesis is $l_\infty$-error. It is defined simply as the size of highest component in vector $\vec{r} - \vec{e}$, consequently

$$\|\vec{r} - \vec{e}\|_\infty = \max_{1 \le j \le M} |r_j - e_j|. \tag{3.62}$$

### Concentration profile preserving

Lets consider the pure advection equation (2.15). Since there is no other part that can influence the mass movement throughout the volume than advection the initial profile shape has to be preserved for all $t > 0$ (assuming constant uniform velocity field). Therefore the solution of pure advection equation can be viewed as shifted initial profile of concentration in space. From this point of view, the following different error measures are often taken into account.

The first measure from this group is the *relative root mean squared error* (RMSE). The differences between the exact and calculated solution are normalized by the difference between peak and minimum concentration levels. The result is the number bigger or equal to zero where one means 100 percent error with respect to concentration interval among initial peak and initial zero levels. Let $\vec{r}$ be the vector of calculated values and $\vec{e}$ be the vector of exact values then RMSE for one-dimensional equation is calculated as

$$RMSE = \frac{\sqrt{\sum_{j=1}^{M}(r_j - e_j)^2/M}}{Peak_0 - Min_0}, \tag{3.63}$$

where $M$ is number of points in the domain, $r_j$ and $e_j$ are the values in dimension $j$ of vectors $\vec{r}$ and $\vec{e}$, $Peak_0$ and $Min_0$ is peak and minimum concentration in the initial time.

The next error measure is the *peak error* (PeakE) represented by equation of the form

$$PeakE = 1 - \frac{Peak_c - Min_c}{Peak_0 - Min_0}, \tag{3.64}$$

where $Peak_c$ and $Peak_0$ is the calculated and initial peak of the concentration, $Min_c$ and $Min_0$ is the calculated and initial minimum of the concentration level.

The *mass distribution ratio* (DistrE) represents the shape preservation without reference to the advected shape. The algorithm can, e.g., nicely preserve shapes but it shifts the position of the shape to a wrong place. Thus its RMSE error would be relatively high. On the other hand the distribution error would be much smaller. The distribution error for one-dimensional case is defined as

$$DistrE = 1 - \frac{\sum_{j\in\Omega_j} r_j}{\sum_{j\in\Omega_j} e_j}, \tag{3.65}$$

where $\Omega_j$ refers to domain where $r_j$ and $e_j$ differs from $Min_0$.

### Error measures for real experiments

The experiments with the real data consist of only few places where the final concentrations were measured. Therefore, the special measures are taken into account when calculated solution is evaluated using the experimental data [23].

First one is the *normalized mean square error* (NMSE) which represents the quadratic error of the predicted quantities in relation to the observed ones. Best result is indicated by values nearest to 0. NMSE has the following form

$$NMSE = \frac{\overline{(C_o - C_p)^2}}{\overline{C_o}\cdot\overline{C_p}}, \tag{3.66}$$

where $C$ (kg m$^{-3}$) is concentration with subscripts with the meaning of: $o$ – observed, $p$ – predicted. The overbar determines the average of a quantity. The used symbols in equation (3.66) are also used in the following equations.

The second statistical index is represented by the *correlation coefficient* (COR). The correlation is maximal when it reaches 1 and is defined as

$$COR = \frac{\overline{(C_o - \overline{C_o})(C_p - \overline{C_p})}}{\overline{\sigma_o \sigma_p}}, \tag{3.67}$$

where $\sigma_o$ (kg m$^{-3}$) and $\sigma_p$ (kg m$^{-3}$) are standard deviations of observed and predicted quantities.

The *fractional of data* (FA2) states the amount of samples that are within the defined space

$$FA2 \approx 0.5 \leq \frac{C_p}{C_o} \leq 2. \tag{3.68}$$

The *fractional bias* (FB) denotes whether the predicted quantities underestimate or overestimate the observed ones (the closer to 0 the better)

$$FB = \frac{\overline{C_o} - \overline{C_p}}{0.5(\overline{C_o} + \overline{C_p})}. \tag{3.69}$$

The last one is the *fractional standard deviation* (FS) that indicates the statistical precision as a fractional result and again if it is closer to 0 the results are more precise

$$FS = \frac{\sigma_o - \sigma_p}{0.5(\sigma_o + \sigma_p)}. \tag{3.70}$$

## 3.3 Numerical methods for atmospheric ADE solution

In section 3.1 the most common numerical methods and frameworks to solve general PDE systems were introduced. In this section, several widely used concrete methods for solving ADE and directly/indirectly used further in this thesis are presented. A set of experiments with the same conditions is defined here and it is used to basically evaluate the schemes and to compare them with each other.

### 3.3.1 Basic spatial discretization

The numerical solution to ADE is usually defined on final number of points in time-space domain of the described problem. The sections 3.1.1 and 3.1.2 show the approximation of partial derivatives by finite difference quotients. It is assumed there that the finite differences are defined on a uniform space grid, i.e., the distances between neighbouring vertices are always the same.

Let one-dimensional uniform grid $\Omega_h = \{x_1, x_2, ..., x_m\}$ is defined as the set of $m$ grid points $x_j = jh$ where $h = 1/m$. In the following text, the approximation on grid $\Omega_h$ to exact solution $u(x_j, t)$ is marked as $w_j(t)$.

### Advection equation

Lets consider the case of advection equation (2.15) in the form

$$u_t + a u_x = 0, \tag{3.71}$$

where $u_t = \frac{\partial y}{\partial t}$, $u_x = \frac{\partial u}{\partial x}$ and $a$ is advection coefficient.

Forward difference formula (3.7) to approximate partial derivative $u_x$ leads to the *first-order upwind* advection discretization

$$w'_j(t) = \frac{a}{h}\left(w_{j-1}(t) - w_j(t)\right), \quad j = 1, 2, ..., m, \tag{3.72}$$

where $h$ is grid step. The equation (3.72) should be used only for $a > 0$ for stability reasons. First-order upwind scheme for $a < 0$ reads

$$w'_j(t) = \frac{a}{h}\left(w_j(t) - w_{j-1}(t)\right), \quad j = 1, 2, ..., m. \tag{3.73}$$

Similarly, central difference (3.10) can be used to approximate spacial derivative $u_x$ and it leads to the *second-order central* advection discretization

$$w'_j(t) = \frac{a}{2h}\left(w_{j-1}(t) - w_{j+1}(t)\right), \quad j = 1, 2, ..., m. \tag{3.74}$$

In order to compare upwind and central difference schemes simple one-dimensional experiments were done. The results of experiments with smooth initial profile $w(0) = sin(\pi x)^{100}$ in time $t = 1$, number of points $m = 50$ with distance $h = 1/50m$ are shown in figure 3.4, where the black solid line is the exact solution and the grey line is the approximated solution $w$.

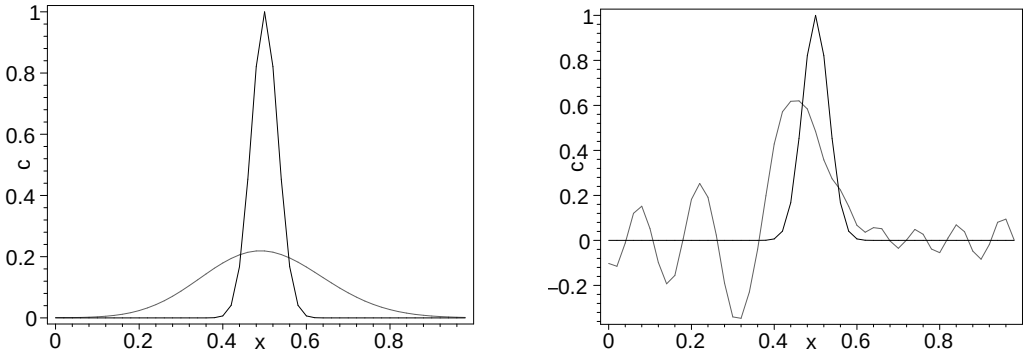



Figure 3.4: Test result of upwind/central $2^{nd}$-order advection schemes.

The second set of experiments was done with the same settings except the initial profile. Here, the sharp box-shape initial profile which is shown together with results in figure 3.5 was used.

During all experiments in this section the numerical $4^{th}$-order Runge-Kutta method (3.17) was used for integration of semi-discrete schemes and its step was chosen reasonably small to have minimal impact on the accuracy.

Since the central difference schemes tend to oscillate over the solution and the first-order upwind scheme brings artificial diffusion to the solution, the upwind scheme of higher order is often preferred in applications. As an example, the *third-order upwind-biased* scheme has the form for $a > 0$ [26]

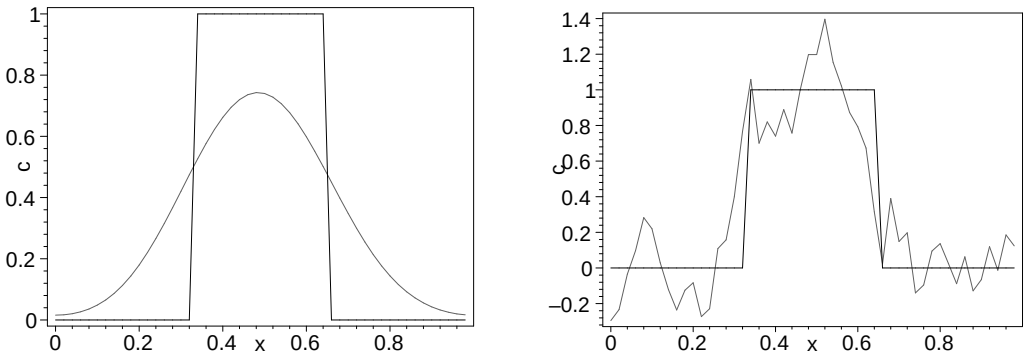



Figure 3.5: Test result of upwind/central $2^{nd}$-order advection schemes for box-shape initial profile.

$$w_j'(t) = \frac{a}{h}\left(-\frac{1}{6}w_{j-2}(t) + w_{j-1}(t) - \frac{1}{2}w_j(t) - \frac{1}{3}w_{j+1}(t)\right), \quad j = 1, 2, ..., m. \tag{3.75}$$

For $a < 0$ the scheme reads

$$w_j'(t) = \frac{a}{h}\left(\frac{1}{3}w_{j-1}(t) + \frac{1}{2}w_j(t) - w_{j+1}(t) + \frac{1}{6}w_{j+2}(t)\right), \quad j = 1, 2, ..., m. \tag{3.76}$$

The experiment with the same inputs and output time $t = 1$ as before was done also for the third order upwind-biased scheme. The results are shown in figure 3.6. There is shown that the profile of initial shape is better preserved than in case of upwind advection scheme, however, non-physical negative concentration values are calculated in this case. Moreover, the result for sharp box-shaped profile shows the noticeable oscillations around the edges.

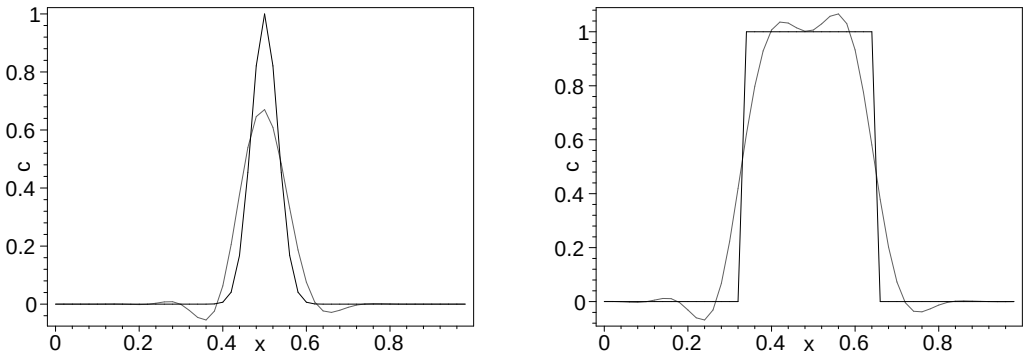



Figure 3.6: Test result of the third order upwind-biased scheme for sine- (left) and box-shape (right) initial profiles.

There exist numerous advection schemes in literature. The comprehensive overview of the higher-order advection schemes and their comparison can be seen in [54]. Quite young WENO profile [35] is also evaluated there and due to its very good properties it will be described and evaluated later in section 3.3.2.

### Diffusion equation

The equation for diffusion process in ADE will be further considered as separated and will have the following form

$$u_t = Du_{xx}, \tag{3.77}$$

where $u_t = \frac{\partial u}{\partial t}$, $u_{xx} = \frac{\partial^2 u}{\partial x^2}$ and $D$ is diffusion coefficient.

Central difference formula (3.10) to approximate partial derivative $u_{xx}$ leads to the *second-order central* diffusion discretization

$$w_j'(t) = \frac{D}{h^2}\left(w_{j-1}(t) - 2w_j(t) - w_{j+1}(t)\right), \quad j = 1, 2, ..., m. \tag{3.78}$$

The second-order central diffusion scheme is very commonly used in diffusion equation as it turned out to be sufficiently accurate. The experiment with the same initial conditions (profile $w(0) = sin(\pi x)^{100}$) was used in order to demonstrate the scheme properties. Left part in figures 3.7 and 3.8 show that even for relatively coarse grid $h = 1/20$ (dotted line) the error of the scheme is relatively small in comparison with exact solution (solid line).

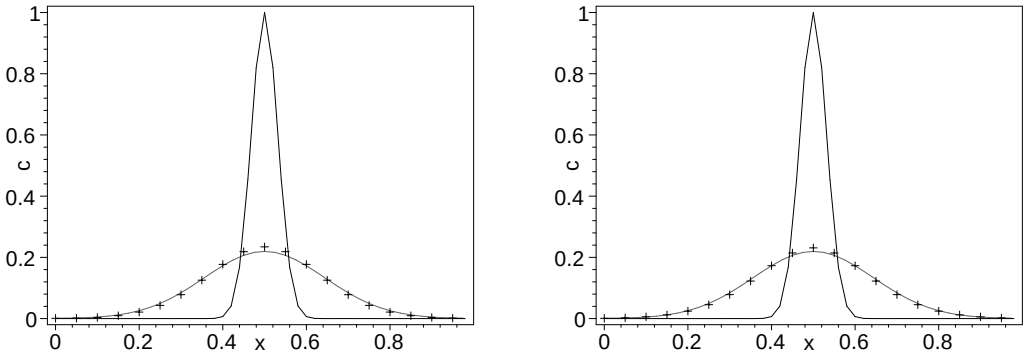



Figure 3.7: Test result of second-order central diffusion (left) and Cranc-Nicolson (right) schemes for sine-shape initial profile.

Next to the second-order central difference, Crank-Nicolson (see equation (3.20)) scheme is often used to solve the diffusion part of ADE. Therefore, the experiments with the same initial conditions, grid resolution and time stepping size were done for the comparison. The results can be seen on the right part in figures 3.7 and 3.8 where it is shown that Crank-Nicolson method reached slightly better accuracy.

### 3.3.2 WENO method

Weighted essentially non-oscillatory (WENO) [35] schemes have been widely used in numerical solutions of hyperbolic partial differential equations (PDEs) and other advection dominated problems. The main advantage of such schemes is their capability to achieve arbitrarily high order formal accuracy in smooth regions while maintaining stable and non-oscillatory discontinuity transitions.

The first WENO scheme was introduced in 1994 in the pioneering paper [35], in which the third-order accurate finite volume WENO scheme was designed. Jiang and Shu [28] provided a general framework to construct arbitrary order accurate finite difference WENO

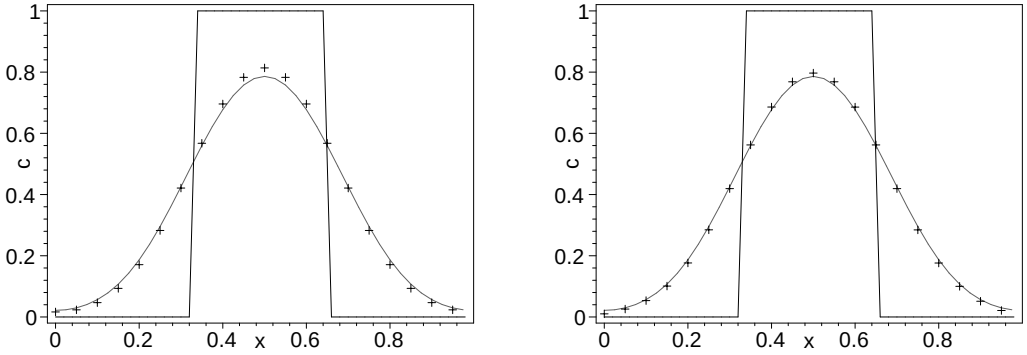



Figure 3.8: Test result of second-order central diffusion (left) and Cranc-Nicolson (right) schemes for box-shape initial profile.

schemes that are more efficient for multi-dimensional calculations. Till now, most of the applications use the fifth-order accurate WENO scheme.

An approximation procedure, not directly related to PDEs, lies at the heart of the WENO schemes. The following simple problem of interpolation is further used in order to describe this approximation procedure. Lets assume a uniform mesh $... < x_1 < x_2 < x_3 < ...$ and the point values of a function $u_i = u(x_i)$. The goal is to find an approximation of the function $u(x)$ at a point other than the nodes $x_i$, for example at the half nodes $x_{i+1/2}$.

The common way to achieve this is to use the traditional approach of interpolation. As an example, a unique polynomial of degree at most two, denoted by $p_1(x)$ that interpolates the function $u(x)$ at the mesh points in the stencil $S_1 = \{x_{i-2}, x_{i-1}, x_i\}$ could be found. One could then use $w^{(1)}_{i+1/2} = p_1(x_{i+1/2})$ as an approximation to the value $u(x_{i+1/2})$ that is third-order accurate

$$w^{(1)}_{i+1/2} - u_{x_{i+1/2}} = \mathcal{O}(\Delta x^3),$$

assuming the function $u(x)$ is smooth in stencil $S_1$. Similarly, if different stencils $S_2 = \{x_{i-1}, x_i, x_{i+1}\}$ and $S_3 = \{x_i, x_{i+1}, x_{i+2}\}$ are chosen, different interpolation polynomials $p_2(x)$ and $p_3(x)$ giving approximations $w^{(2)}_{i+1/2}$ and $w^{(3)}_{i+1/2}$ are obtained. The combination of all three stencils $S_1$, $S_2$ and $S_3$ can be used to find the interpolation polynomial $p(x)$ of degree at most four on big stencil $S$ giving an approximation $w_{i+1/2}$ which is fifth-order accurate [28].

The essential part of the fifth-order WENO procedure is that the fifth-order approximation $w_{i+1/2}$ can be written as a linear convex combination of the three third-order approximations $w^{(1)}_{i+1/2}$, $w^{(2)}_{i+1/2}$ and $w^{(3)}_{i+1/2}$ as

$$w_{i+1/2} = \gamma_1 w^{(1)}_{i+1/2} + \gamma_2 w^{(2)}_{i+1/2} + \gamma_3 w^{(3)}_{i+1/2}, \tag{3.79}$$

where the constants $\gamma_1$, $\gamma_2$ and $\gamma_3$, satisfying $\gamma_1 + \gamma_2 + \gamma_3 = 1$, are usually referred to as the *linear weights*.

The linear combination as defined in equation (3.79) expects that approximated function $u(x)$ is smooth at each used stencil. However, this is not always true and non-linear weights $\mu_j$ are used instead

$$w_{i+1/2} = \mu_1 w^{(1)}_{i+1/2} + \mu_2 w^{(2)}_{i+1/2} + \mu_3 w^{(3)}_{i+1/2}, \tag{3.80}$$

where $\mu_j \geq 0$, $\mu_1 + \mu_2 + \mu_3 = 1$. The idea now is to choose $\mu_j \approx \gamma_j$ if $u(x)$ is smooth in big stencil $S$ and $\mu_j \approx 0$ otherwise. The choice of the non-linear weights $\mu_j$ relies on the *smoothness indicators* $\beta_j$ whose definition can be found in [50]. Using these smoothness indicators, the non-linear weights can be defined as

$$\mu_j = \frac{\widetilde{\mu}_j}{\widetilde{\mu}_1 + \widetilde{\mu}_2 + \widetilde{\mu}_3}, \quad \text{where } \widetilde{\mu}_j = \frac{\gamma_j}{(\epsilon + \beta_j)^2}. \tag{3.81}$$

Here $\epsilon$ is a small positive number to avoid the denominator to become zero.

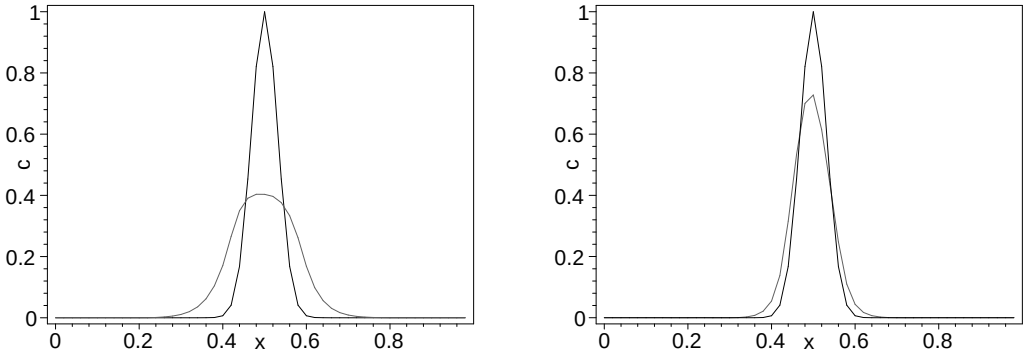



Figure 3.9: Test result of WENO $3^{rd}$- and $5^{th}$-order schemes with sine-shape initial conditions.

In order to evaluate the WENO scheme, its third- and fifth-order variants were implemented and tested using the same conditions like other experiments in this section. Figure 3.9 shows the results of the experiment with sine-shaped initial conditions. It can be clearly seen that $3^{rd}$-order WENO has a tendency to smooth the solution too much. The fifth-order variant behaves much better in this test and the resulting profile shape is the most accurate in all experiments made so far in this section.

The second set of experiments was done with the box-shaped initial profile. As it is shown in figure 3.10, much better results are obtained in the fifth-order scheme and again it outperforms all other schemes described before.

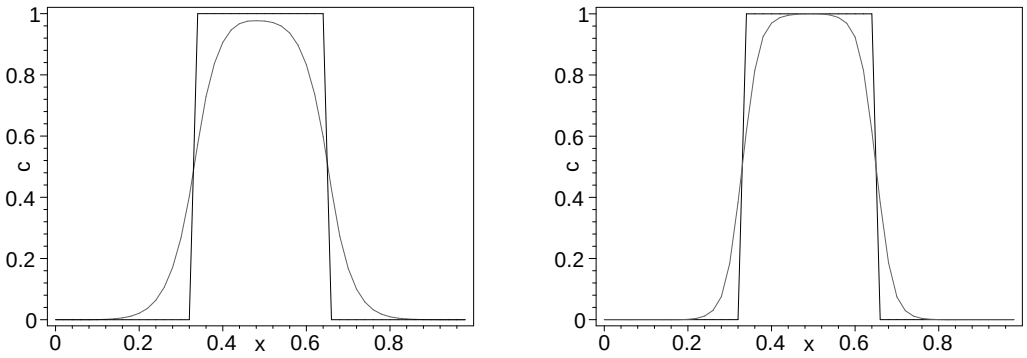



Figure 3.10: Test result of WENO $3^{rd}$ and $5^{th}$ order schemes with box shape initial conditions.

From WENO definition, its first-order accurate variant is equivalent to upwind scheme. Figure 3.11 shows the results of the experiment with box-shaped initial profile and the perfect result equivalence between first-order WENO and upwind schemes.

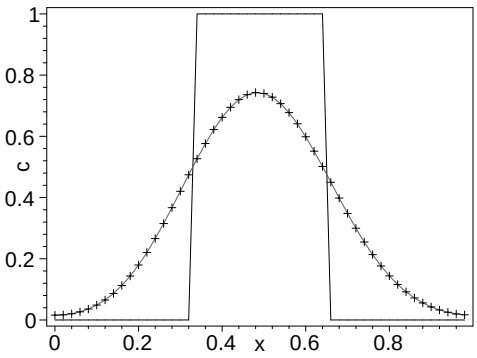



Figure 3.11: Test result of WENO $1^{st}$-order (point graph) and numerically equivalent upwind (line graph) schemes.

The very good properties and results of WENO schemes showed in this thesis were also confirmed in comprehensive review paper on different advection schemes that are commonly used [54]. The negative aspect on WENO procedure, also shown there, is its computational cost which is several times higher than in case of common high-order schemes based on Taylor-series expansion. However, the stability, monotonicity and non-oscillatory properties often outclass its relative slowness.

### 3.3.3 Walcek method

Many schemes dedicated not only to atmospheric advection have been proposed during last decades. The well known and widely used is the scheme proposed by Bott [5] on whose basis quite new and very accurate scheme was designed by Walcek et al. [58].

The method uses rather the *mixing ratios* instead of concentrations in order to allow model the scenarios with variable density of the fluid. The mixing ratio $R$ (kg kg$^{-1}$) of a tracer is defined as ratio between mass or concentration $C$ (kg m$^{-3}$) of a tracer and density $\rho$ (kg m$^{-3}$) of the fluid

$$R \equiv \frac{C}{\rho}.$$

To describe the algorithm, lets further assume one-dimensional advection equation and uniform grid. The initial guess of the mixing ratio $R_i^{guess}$ in grid cell $i$ at time $t + \Delta t$ is obtained as

$$R_i^{guess} = \left( R_i^t D_{d-1} - \frac{F_{i+1/2}}{\Delta x_i} + \frac{F_{i-1/2}}{\Delta x_i} \right) / D_d, \tag{3.82}$$

where $F_{i\pm1/2}$ (kg m$^{-2}$) are fluxes of tracer across the cell boundaries $i \pm 1/2$ and $D_d$ (kg m$^{-3}$) are dimensional dependent fluid densities. For one-dimensional calculations in incompressible fluids, $D_{d-1} = D_d = 1$. For multi-dimensional calculations in incompressible fluids, $D_{d-1} = 1$ and $D_d = 1 - \Delta t \Delta a_i / \Delta x_i$, where $\Delta a_i$ (m s$^{-1}$) is relative wind speed in cell $i$.

Fluxes and velocities are defined at the edges (faces) of the grid cells where the mixing ratios are defined. Fluxes at cell faces are defined by introducing an *outflowing* mixing ratio $R_f$ as

$$F_{i+1/2} = (\rho_0 a)_{i+1/2} \Delta t R_f, \tag{3.83}$$

where $(\rho_0 a)_{i+1/2}$ is mass flux across the cell boundary $i + 1/2$, defined using the initial fluid density $\rho_0$, $\Delta t$ is integration time step and $a$ is advection variable. The initial density $\rho_0$ can be defined using upwinding as one of the cell boundary densities or as average of edge densities in the cell.

$R_f$ is the average mixing ratio in the fluid that is advected into the neighbouring grid cell. The definition of $R_f$ allows to limit the fluxes in reasonable physical values. The initial guess $R_f^*$ of $R_f$ can be algebraically derived as a Courant number-dependent linear combination of the mixing ratios in the three cells closest to the cell face where fluxes are calculated [58]. The linear combination includes the *sharpening factor* $\alpha$ which can be adjusted to actual conditions.

The initial guess $R_f^*$ can produce unrealistic estimates of mixing ratios at cell boundaries (local overshoot or undershoot). It is therefore reasonable to define the physical boundaries of the fluxes and thus to limit the fluxes as

$$R_{min} = \min(R_i, R_{i+1}) \leq R_f \leq \max(R_i, R_{i+1}) = R_{max}. \tag{3.84}$$

The method introduces also mixing ratio limiting at the end of step calculation. The idea is the following. As long as not all the fluid in one grid is replaced in one time step (Courant number $< 1$), at time $t + \Delta t$, it is physically impossible for mixing ratios to be greater than the highest mixing ratio $R_{max}^{t+\Delta t}$ or lower than the lowest mixing ratio $R_{min}^{t+\Delta t}$ of the upwind cell or the mixing ratio of the cell initially. Such mixing ratio limitations can be expressed as

$$R_i^{t+\Delta t} = \max[\min(R_{max}^{t+\Delta t}, R_i^{guess}), R_{min}^{t+\Delta t}], \tag{3.85}$$

where $R_i^{guess}$ is equal to $R_f^*$ limited by equation (3.84). When mixing ratios are limited the appropriate fluxes at upwind cell boundaries need to be recalculated for the next step.

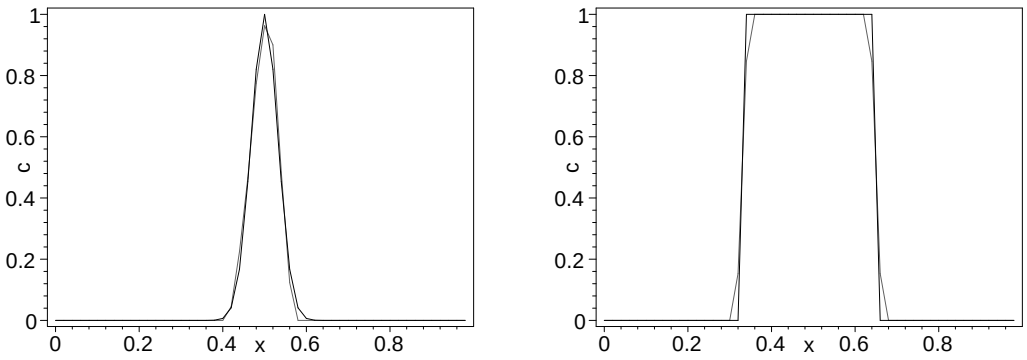



Figure 3.12: Test result of Walcek scheme with sine- and box-shape initial conditions.

Similarly to other schemes presented in this section, Walcek method was also tested using the same conditions as defined before. The results, shown in figure 3.12, confirm that Walcek method has the excellent properties and it outperforms the other schemes described before.

### 3.3.4 ELLAM framework

The previous subsections showed the commonly used specialized methods for advection equations. On the other hand, ELLAM framework belongs to family of localized adjoint methods (LAM) which are based on philosophy of algebraic theory of numerical methods by Herrera [24]. In LAM, the weighted or testing functions $g(\vec{x})$ are defined and they are used for definition of weak form of governing partial differential equation. Let $\mathcal{L}$ is the differential operator, then the governing equation is defined as

$$\mathcal{L}u(\vec{x}) = f(\vec{x}), \quad \vec{x} \in \Omega,$$

where $u$ is dependent variable and $\vec{x}$ is vector of independent variables in $\Omega$ space. The weak form of governing equation has the form

$$\int_{\Omega} (\mathcal{L}u)\, g(\vec{x}) d\vec{x} = \int_{\Omega} f\,(\vec{x})\, g(\vec{x}) d\vec{x}. \tag{3.86}$$

In general LAM, the space $\Omega$ is discretized into a certain number of intervals or elements $\Omega_i\,(i = 1, 2, ..., E)$. Equation (3.86) is then defined as a sum of inner and outer integrals of each element. The inner integrals include the integrated function consisting of $\mathcal{L}^* g$ expression whereas $\mathcal{L}^*$ is the operator adjoined to $\mathcal{L}$.

The procedure in LAM then defines the testing functions $g$ in a way such that they satisfy the homogeneous adjoint equation and thus $\mathcal{L}^* g = 0$. Then all the inner integrals in the elements will be zero and it suffices to evaluate the outer integrals. Despite of a way of discretization, the choice of testing functions is essential in LAM procedure and it is influenced by the actual problem.

In order to demonstrate ELLAM procedure, lets assume one-dimensional ADE equation defined as

$$\mathcal{L}u = \frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} - D\frac{\partial^2 u}{\partial x^2} = f(x,t), \quad 0 < x < L, \quad t > 0, \tag{3.87}$$

where $a$ (m s$^{-1}$) is wind speed, $D$ (m$^2$ s$^{-1}$) is diffusion coefficient, $u$ (kg m$^{-3}$) represents solution and $L \in \mathcal{R}$ (m).

Following the definition (3.86), the weak form of equation (3.87) has the form

$$\int_{\Omega_t}\int_{\Omega_x} \left(\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} - D\frac{\partial^2 u}{\partial x^2}\right) g(x,t) dx dt = \int_{\Omega_t}\int_{\Omega_x} f(x,t) g(x,t) dx dt. \tag{3.88}$$

The adjoint operator with respect to $\mathcal{L}$ of equation (3.87) is equal to zero and has the form [10]

$$\mathcal{L}^* g = -\frac{\partial g}{\partial t} - a\frac{\partial g}{\partial x} - D\frac{\partial^2 g}{\partial x^2} = 0. \tag{3.89}$$

Now the answer to question why ELLAM is labelled as Eulerian-Lagrangian method comes. The space of solutions to equation (3.89) is infinite. The different choice of testing functions leads to different classes of approximations including FEM and method of characteristics [10]. In ELLAM, the adjoint operator in equation (3.89) is divided into two parts which are equalled to zero

$$-\frac{\partial g}{\partial t} - a\frac{\partial g}{\partial x} = 0, \tag{3.90}$$

$$-D\frac{\partial^2 g}{\partial x^2} = 0. \tag{3.91}$$

Equation (3.90) guarantees that function $g$ is constant along the *characteristic curve* defined as $\frac{dx}{dt} = a$ [46] (Lagrangian part) and equation (3.91) defines the function $g$ as linear on $x$ (Eulerian part). Function $g$ is defined for each grid point with index $i$ for time $t_n$, denoted as $g_i^n(x,t)$, and it can have the following form

$$g_i^n(x,t) = \begin{cases} \frac{x-x_{i-1}}{\Delta x} + a\frac{t_n - t}{\Delta x} & x_{i-1}^* \leq x \leq x_i^*, t_{n-1} < t < t_n \\ \frac{x_{i+1}-x}{\Delta x} + a\frac{t_n - t}{\Delta x} & x_i^* \leq x \leq x_{i+1}^*, t_{n-1} < t < t_n \\ 0 & \text{for other } x, t, \end{cases} \tag{3.92}$$

where $x_{i-1}^*$, $x_i^*$ and $x_{i+1}^*$ are points at time $t_{n-1}$ corresponding to $x_{i-1}$, $x_i$ and $x_{i+1}$ points in time $t_n$ lying on a characteristic curve.

The integration by parts of equation (3.88) and the application of conditions (3.90, 3.91) leads to the following form [46]

$$\begin{aligned} \int_{\Omega_x} u(x,t_n) g_i^n(x,t_n) dx - \int_{\Omega_x} u(x,t_{n-1}) g_i^n(x,t_{n-1}) dx \\ + D\int_{t_{n-1}}^{t_n}\int_{\Omega_x} \frac{\partial u}{\partial x}\frac{\partial g_i^n}{\partial x} dx dt \\ = \int_{\Omega_t}\int_{\Omega_x} f(x,t) g_i^n(x,t) dx dt, \end{aligned} \tag{3.93}$$

where it is assumed that the testing functions $g_i^n(x,t)$ are zero outside the domain $\Omega_x$ for each $t \in [t_{n-1}, t_n]$. Equation (3.93) is defined for each element $i$ and the overall solution is a sum of all the solutions of particular elements. This leads to the process where the set of algebraic equations is solved at each time step. The details about the used form of ELLAM in this thesis are presented in section 4.2.

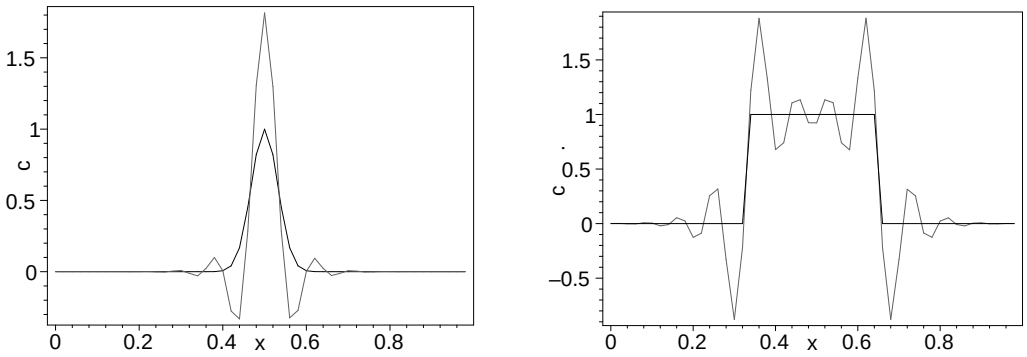



Figure 3.13: Test result of ELLAM scheme with sine- and box-shape initial conditions with fine time stepping resulting in big oscillations.

The advantage of ELLAM framework is a natural treatment of boundary conditions in its formulation. When the boundary conditions are applied to equation (3.88) and then integrated by parts, the equation similar to (3.93) arises with separated integrals for the actual boundary conditions.

The simple experiments were done to compare the ELLAM method with the rest of the schemes in this section using the same conditions as in previous subsections (diffusion coefficient $D$ was set to zero). The results for sine- and box-shaped initial profiles are shown in figure 3.13. It is known that ELLAM in its general form suffers by quite big oscillations when used for pure advection problems using many time steps which is confirmed by the tests also here. The second set of tests was done with time steps five times bigger than in other cases in this section. Their results with significantly smaller oscillations are shown in figure 3.14.

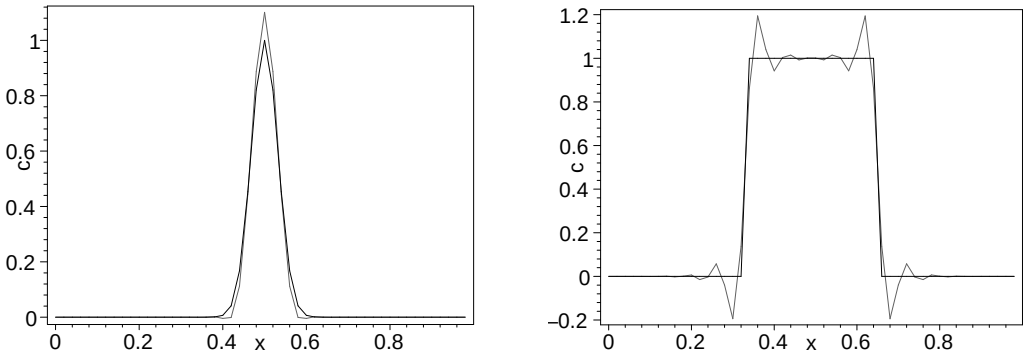



Figure 3.14: Test result of ELLAM scheme with sine- and box-shape initial conditions using quintuple integration step.

In spite of the problems of oscillations it has been shown that ELLAM outperformed many other advection schemes due to its stability and accuracy [59] and thus it will be in main focus further in the thesis.

### 3.3.5 ADE discretization in AURORA

AURORA [44] is the urban scale air quality model which includes terrain, emission, chemistry and advection-diffusion modules. The latter forms the heart of the overall model and thus the stable and accurate implementation is essential for the actual predictions.

Advection-diffusion equation used in AURORA module has the following form [44]

$$\frac{\partial \rho R}{\partial t} = -\nabla \cdot (\rho \vec{a} R) + \frac{\partial}{\partial z}(\rho D \frac{\partial R}{\partial z}) + S_V(x, y, z), \tag{3.94}$$

where $R$ (kg kg$^{-1}$) is tracer mixing ratio, $\rho$ (kg m$^{-3}$) is air density, $\vec{a}$ (m s$^{-1}$) is wind vector, $D$ (m$^2$ s$^{-1}$) is turbulent diffusion coefficient and $S_V$ (kg m$^{-3}$ s$^{-1}$) is volume source term. The advection and diffusion parts are treated separately using operator splitting technique and the specialized numerical schemes are used for both.

#### Advection

The advection part in AURORA's ADE module is calculated using Walcek method (see section 3.3.3). The difference from that method is in definition of fluxes at the cell interfaces. The fluxes are estimated using the second-order Van Leer scheme

$$R_{f,i+1/2} = R_i + \frac{(R_{i+1} - R_{i-1})(1-c)}{4}\alpha \quad \text{for } a_{i+1/2} \geq 0 \tag{3.95}$$

$$R_{f,i-1/2} = R_i + \frac{(R_{i-1} - R_{i+1})(1-c)}{4}\alpha \quad \text{for } a_{i-1/2} < 0, \tag{3.96}$$

where $c = a\Delta t/\Delta x < 1$ is Courant number and $\alpha$ is sharpening factor to maintain concentration gradients during advection. The flux limiting techniques are further used to achieve the desirable monotonicity properties of the scheme. The results of one-dimensional experiments shown in [44] correspond to the ones performed in this section.

### Vertical diffusion with source term

The diffusive part of equation (3.94) can be written as

$$\frac{\partial \rho R}{\partial t} = \frac{\partial}{\partial z}(\rho D \frac{\partial R}{\partial z}) + S(x,y,z). \tag{3.97}$$

Here, the volume sources denoted by $S(x,y,z)$ are represented in the model as the set of local source points even for linear sources of pollution such as road traffic.

The equation (3.97) is discretized using the following scheme [44]

$$\frac{w_j(t+\Delta t) - w_j(t)}{\Delta t} = -\frac{1}{\rho_j}\frac{F_{j+1} - F_j}{z_{j+1} - z_j} + \frac{1}{\rho_j}S_j, \tag{3.98}$$

where $S_j$ (kg m$^{-3}$ s$^{-1}$) is volume source term and $F$ (kg m$^{-2}$ s$^{-1}$) is diffusive flux defined as

$$F_j = -(\rho_j D_j)\frac{w_j - w_{j-1}}{\frac{1}{2}(z_{j+1} - z_{j-1})}. \tag{3.99}$$

Here, $D_j$ is diffusion coefficient at node $j$. Putting $w_j = \frac{1}{2}(w_j(t) + w_j(t+\Delta t))$ and assuming continuous point source ($S$ is constant), constant air density ($\rho = 1$ kg m$^{-3}$) and uniform grid ($z_{j+1} - z_j = \Delta z$ for all $j$) lead to the following form of set of equations

$$\begin{aligned} -\beta D_j w_{j-1}(t+\Delta t) + [1 + \beta(D_j + D_{j+1})]w_j(t+\Delta t) - \beta D_{j+1} w_{j+1}(t+\Delta t) = \\ \beta D_j w_{j-1}(t) + [1 - \beta(D_j + D_{j+1})]w_j(t) + \beta D_{j+1} w_{j+1}(t) + S_j \Delta t, \end{aligned} \tag{3.100}$$

where $\beta = \frac{\Delta t}{(\Delta z)^2}$. Crank-Nicolson discretization procedure leading to equation (3.100) represents the set of equations which, from matrix point of view, is in tridiagonal form and can be solved efficiently.

The two simple experiments were done in order to demonstrate the performance of the used unconditionally stable implicit scheme. The conditions of the one-dimensional experiments were the following. The mixing level height of the vertical diffusion model was set to $H = 1500$ m. At time $t = 0$ s the level of concentration was set to zero everywhere. The vertical diffusion model was chosen with parabolic shape defined as

$$D(z) = K a_\star z \left(1 - \frac{z}{H}\right),$$

where $K$ is von Karman constant (set to 0.4) and $a_\star$ is friction velocity (set to 1 ms$^{-1}$). At time $t = 0$ s the continuous ground level source term was switched on with strength of

$10^{-6}$ kg m$^{-3}$s$^{-1}$. The time stepping was used with 20 subsequent time steps $\Delta t = 1800$ s which is 10 hours together. The analytical solution for the case is known [44] and was also used here as a reference solution.

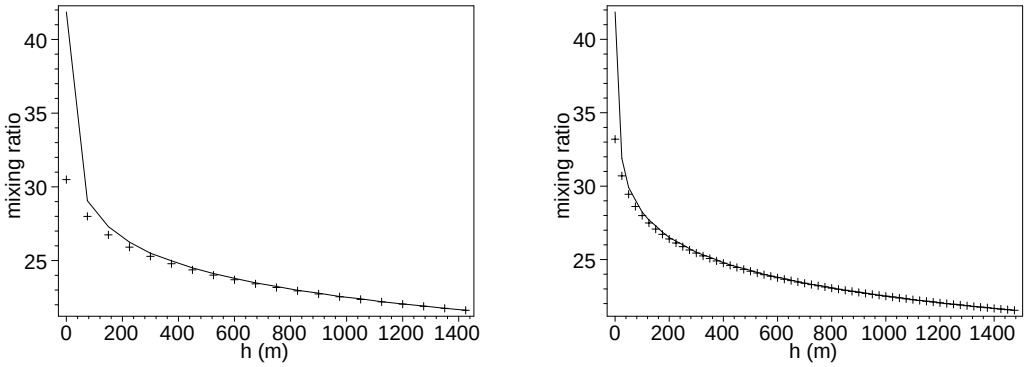



Figure 3.15: Test result of diffusion method used in AURORA with the two different grid spacing, $\Delta z = 75$ m (left) and $\Delta z = 25$ m (right).

The results of the performed experiments are shown in figure 3.15. Here the two cases show quite good agreement with analytical solution. The difficulties can be seen in case of 75 m grid spacing. This is reasonable since bigger discontinuities caused by source term appear near the ground. The spacing of 25 m performed very well which corresponds with results presented in [44].

### 3.3.6 Summary

Various schemes for advection and diffusion equation were introduced in this section. The aim of the description of the schemes was not only to describe existing and widely used schemes but also to compare them with each other using simple test cases. In this subsection the comparison of the schemes is finalized by presenting their accuracy using $L_n$-norm error measures defined in section 3.2.6.

#### Advection equation

In many practical applications, the model represented by ADE equations is divided into its advection and diffusion parts which are solved separately. Much more numerical problems are caused when dealing with advection equations, thus more schemes were presented and evaluated with this respect.

The traditional schemes derived from Taylor series expansion have some troubles when solving the advection equation with steep or sharp initial profiles. Upwind scheme adds the artificial diffusion to its calculation and thus the initial profile in pure advection problem is not preserved at all. On the other hand, second-order central difference scheme preserves the initial profile a bit better, however, its solution contains large oscillations around the exact solution and even global overshoots and undershoots.

The monotonicity preserving basic upwind scheme was extended and the higher order upwind schemes were developed. The third-order upwind scheme is presented in this section and it shows more accuracy than the basic variant, however, in cost of non-monotonic behaviour. Therefore, the flux-limiting technique was used in the scheme which leads to monotonicity preserving, although a bit less accurate, third-order upwind limited scheme.

| Scheme | $L_1$-norm error | $L_2$-norm error | $L_\infty$-norm error |
|---|---|---|---|
| Upwind | $0.990 \cdot 10^{-1}$ | $0.347 \cdot 10^{-1}$ | 0.781 |
| Second-order central | 0.140 | $0.392 \cdot 10^{-1}$ | 0.530 |
| Upwind third-order biased | $0.382 \cdot 10^{-1}$ | $0.655 \cdot 10^{-2}$ | 0.330 |
| Upwind third-order biased limited | $0.405 \cdot 10^{-1}$ | $0.102 \cdot 10^{-1}$ | 0.452 |
| WENO first-order | $0.990 \cdot 10^{-1}$ | $0.347 \cdot 10^{-1}$ | 0.781 |
| WENO third-order | $0.631 \cdot 10^{-1}$ | $0.193 \cdot 10^{-1}$ | 0.597 |
| WENO fifth-order | $0.247 \cdot 10^{-1}$ | $0.370 \cdot 10^{-2}$ | 0.272 |
| Walcek | $0.736 \cdot 10^{-2}$ | $0.361 \cdot 10^{-3}$ | $0.792 \cdot 10^{-1}$ |
| ELLAM | $0.818 \cdot 10^{-1}$ | $0.379 \cdot 10^{-1}$ | 0.815 |
| ELLAM (quintuple step) | $0.944 \cdot 10^{-2}$ | $0.582 \cdot 10^{-3}$ | 0.101 |

Table 3.1: $L_n$-norm error measures of performed one-dimensional advection experiments with sine initial profile.

Next to traditional schemes, more sophisticated and complicated schemes were also described in this section. Recently, WENO schemes were introduced and they became very popular when dealing with PDE and general interpolation of functions. Three variants of WENO were evaluated. It was shown that upwind basic scheme is a special case of first-order WENO scheme which is also confirmed by perfect error measurement match in tables 3.1 and 3.2. The fith-order WENO scheme shows that it is more accurate than the schemes shown before and thus it could be a good candidate to solve advection part of ADE equation.

| Scheme | $L_1$-norm error | $L_2$-norm error | $L_\infty$-norm error |
|---|---|---|---|
| Upwind | 0.223 | $0.729 \cdot 10^{-1}$ | 0.498 |
| Second-order central | 0.184 | $0.563 \cdot 10^{-1}$ | 0.767 |
| Upwind third-order biased | $0.673 \cdot 10^{-1}$ | $0.173 \cdot 10^{-1}$ | 0.419 |
| Upwind third-order biased limited | $0.568 \cdot 10^{-1}$ | $0.168 \cdot 10^{-1}$ | 0.402 |
| WENO first-order | 0.223 | $0.729 \cdot 10^{-1}$ | 0.498 |
| WENO third-order | $0.872 \cdot 10^{-1}$ | $0.230 \cdot 10^{-1}$ | 0.427 |
| WENO fifth-order | $0.552 \cdot 10^{-1}$ | $0.150 \cdot 10^{-2}$ | 0.383 |
| Walcek | $0.122 \cdot 10^{-1}$ | $0.188 \cdot 10^{-2}$ | 0.153 |
| ELLAM | 0.187 | $0.906 \cdot 10^{-1}$ | 0.882 |
| ELLAM (quintuple step) | $0.366 \cdot 10^{-1}$ | $0.496 \cdot 10^{-2}$ | 0.195 |

Table 3.2: $L_n$-norm error measures of performed one-dimensional advection experiments with box-shaped initial profile.

The second specialized scheme also evaluated in this section is Walcek method. This is more specialized method designed for multi-dimensional advection equations and possibly to non-uniform grids. The method is monotonicity preserving and it tries to follow discontinuities and steep gradients as much as possible. Walcek scheme showed the excellent behavior in the simple one-dimensional experiments and it outperformed the other tested schemes, including WENO.

The last presented method is ELLAM, which is very specialized and complicated framework specially designed to solve ADE equations. Although, the basic form of ELLAM shows

worse results than Walcek scheme it has an advantage that it can solve the whole ADE equation and it was shown that almost arbitrarily large time steps can be used due to its unconditional stability. The bigger number of time steps used in the method leads to quite large oscillations around the steep gradients.

| Scheme | $L_1$-norm error | $L_2$-norm error | $L_\infty$-norm error |
|---|---|---|---|
| | Sine initial profile | | |
| Second-order central | $0.475 \cdot 10^{-2}$ | $0.382 \cdot 10^{-4}$ | $0.150 \cdot 10^{-1}$ |
| Crank-Nicolson | $0.318 \cdot 10^{-2}$ | $0.211 \cdot 10^{-4}$ | $0.116 \cdot 10^{-1}$ |
| | Box-shaped initial profile | | |
| Second-order central | $0.131 \cdot 10^{-1}$ | $0.220 \cdot 10^{-3}$ | $0.276 \cdot 10^{-1}$ |
| Crank-Nicolson | $0.768 \cdot 10^{-2}$ | $0.683 \cdot 10^{-4}$ | $0.113 \cdot 10^{-1}$ |
| | Continuous source test | | |
| Crank-Nicolson ($d_z = 75$ m) | $0.172 \cdot 10^{-1}$ | $0.374 \cdot 10^{-2}$ | 0.272 |
| Crank-Nicolson ($d_z = 25$ m) | $0.523 \cdot 10^{-2}$ | $0.734 \cdot 10^{-3}$ | 0.207 |

Table 3.3: $L_n$-norm error measures of performed one-dimensional diffusion experiments.

### Diffusion equation

The diffusion part of ADE equation is not so problematic in a sense of numerical computations. The traditional second-order central difference scheme can be used quite successfully for this equation as it is shown in table 3.3 where the errors are of order two or higher.

It was also shown that the usage of unconditionally stable Crank-Nicolson as proposed in AURORA ADE module is actually a good idea. It has slightly better accuracy and if the discretization leads to solve set of equations which are in tri-diagonal form the calculations of this implicit scheme is not very time consuming.

Crank-Nicolson discretization was also successfully used in this section for the first time with non-linear model of the diffusion and it showed very good accuracy when compared to known analytical solution even with presented continuous source term.

# Chapter 4

# Study of numerical solutions of atmospheric ADE equations

The main classification of the concrete area of the pollutant dispersion modeling used in this thesis was described in chapter 2 and the state of the art methods were described and evaluated in chapter 3. Now, the own study of the numerical solutions will be described here using two numerical approaches to solve concrete models.

The first approach uses very popular method of lines (MoL) which general approach is described in section 3.1.2. MoL approach design was used for two steady state and one time-dependent models. The second numerical scheme is ELLAM, which is not frequently used for air-pollution models. The very general description of ELLAM framework is described in section 3.3.4 and on its basis a concrete design will be introduced.

## 4.1 Method of lines utilization

There exist huge number of air pollution models that were designed and evaluated in past decades. The general form of the equation describing atmospheric dispersion can be expressed as follows [7]

$$\frac{\partial C}{\partial t} + \nabla C\vec{a} = \nabla\left(\vec{D}\nabla C\right) + chemistry + emissions + dry\_deposition + wet\_deposition, \tag{4.1}$$

where $C$ (kg m$^{-3}$) is pollution concentration, $\vec{a}$ (m s$^{-1}$) is wind velocity field and $\vec{D}$ (m$^2$ s$^{-1}$) is diffusion vector. The chemistry term presents atmospheric chemistry term that is used for the determination of a chemical substance influence to the atmosphere and to the dispersion process itself. The emissions term expresses the rate of the emissions in the atmosphere and its relation to the atmospheric dispersion of the specific pollutant. The last two terms, dry and wet depositions, are the major sink terms in the model and besides they determine the pollutant behaviour above the terrain surface.

If the chemistry, emissions and wet deposition terms are neglected in equation (4.1) then the following ADE with deposition term is formed

$$\frac{\partial C}{\partial t} + \nabla C\vec{a} = \nabla\left(\vec{D}\nabla C\right) + W\frac{\partial C}{\partial z}, \tag{4.2}$$

where $W$ (m s$^{-1}$) is pollutant gravitational settling velocity.

Equation (4.2) can be furthermore simplified considering the following assumptions. When the wind speed value is sufficiently large, a diffusive transport is negligible in wind direction with respect to advection [18]. Moreover, the coefficients $D_y$ and $D_z$ depend on the downwind distance $x$ only and they are therefore independent on the crosswind distance $y$ and height distance $z$. From these facts, the diffusive terms can be simplified - the brackets are not needed any more and the second derivatives appear. Last assumption is the presence of stationary source with constant strength during time. Therefore, the result of the simplification is the steady state form (2.25). The equation is also shown here and rewritten into $x, y, z$ axes labels for clarity

$$a_x \frac{\partial C}{\partial x} = D_y(x)\frac{\partial^2 C}{\partial y^2} + D_z(x)\frac{\partial^2 C}{\partial z^2} + W\frac{\partial C}{\partial z}, \tag{4.3}$$

where $a_x$ is wind speed along $x$ axis. The complete description of the problem needs boundary conditions to be specified. The first one follows from an assumption of continuous point source with constant strength located in $(0, 0, h_s)$ coordinates

$$C(0, y, z) = \frac{Q}{a_x}\delta(y)\delta(z - h_s), \tag{4.4}$$

where $Q$ (kg s$^{-1}$) is source strength and $\delta$ is Dirac function. The ground boundary condition comes from fact that pollutant deposition onto the ground occurs at a rate proportional to local air concentration [18] (the flat ground is taken into account here for simplicity)

$$\left[D_z(\infty)\frac{\partial C}{\partial z} + WC\right]_{z=0} = [vC]_{z=0}\,. \tag{4.5}$$

Here, deposition velocity $v$ (m s$^{-1}$) depends on many factors such as type and size of pollutant particles, the terrain roughness and the meteorological conditions. The last three boundary conditions follow from natural assumption that pollutant concentration approaches zero far from the source in lateral $y$ directions and high above the ground

$$C(x, +\infty, z) = 0, \tag{4.6}$$
$$C(x, -\infty, z) = 0, \tag{4.7}$$
$$C(x, y, +\infty) = 0. \tag{4.8}$$

### 4.1.1 Ermak's model solution

Following the MoL approach the given ADE (4.3) has to be transformed into the system of ordinary differential equations (ODEs) and then solved using the suitable numerical integration method. All variables except one have to be discretized. Following the results from evaluation of diffusion equation the central differences were used to approximate the diffusive terms. Further the $x$ variable was left to be continuous because of the wind direction. The obtained system of ODEs, after division by $a_x$, is in the form

$$\begin{aligned}\frac{\partial C(x,j,k)}{\partial x} =& \frac{D_y(x)}{a_x} \cdot \frac{C(x,j+1,k) - 2C(x,j,k) + C(x,j-1,k)}{\Delta y^2} + \\ & \frac{D_z(x)}{a_x} \cdot \frac{C(x,j,k+1) - 2C(x,j,k) + C(x,j,k-1)}{\Delta z^2} + \\ & \frac{W}{a_x} \cdot \frac{C(x,j,k+1) - C(x,j,k-1)}{2\Delta z}.\end{aligned} \tag{4.9}$$

The boundary condition (4.4) is transformed into appropriate initial condition

$$\begin{aligned}C(0,0,h_s) &= \frac{Q}{a_x \Delta y \Delta z} \\ C(0,j,k) &= 0 \quad \text{otherwise.}\end{aligned} \tag{4.10}$$

The new boundary conditions in lateral and longitudinal directions are as follows

$$C(x,+N_j,k) = 0, \tag{4.11}$$
$$C(x,-N_j,k) = 0, \tag{4.12}$$
$$C(x,j,+N_k) = 0, \tag{4.13}$$

where $\pm N_j$ are indexes on lateral boundaries and $N_k$ is index on top domain boundary. The last boundary condition that describes the pollutant behaviour on the ground is transformed using the central differences to the form

$$C(x,j,0) = \frac{D_z(\infty)}{2\Delta z(v - W) + D_z(\infty)} C(x,j,2). \tag{4.14}$$

The ODE definition is now finished and suitable numerical integration method can be used such as ones described in section 3.1.2.

### Time-dependent variant

The simple steady state equation (4.3) can be easily extended to its time-dependent variant by adding time derivative $\frac{\partial C}{\partial t}$ to the equation. The still simple variant, used further in the thesis for stability analysis and parallel version evaluation, has the form

$$\frac{\partial C}{\partial t} = -a_x \frac{\partial C}{\partial x} + D_y(x) \frac{\partial^2 C}{\partial y^2} + D_z(x) \frac{\partial^2 C}{\partial z^2} + W \frac{\partial C}{\partial z}. \tag{4.15}$$

The boundary condition representing the source is expressed as

$$C(t,0,y,z) = \frac{Q}{a_x} \delta(y) \delta(z - H). \tag{4.16}$$

The ground boundary condition, for $z = 0$ m, has the same form as before (4.5). Finally, the lateral and longitudinal boundary conditions are

$$C(t,x,+\infty,z) = 0, \tag{4.17}$$
$$C(t,x,-\infty,z) = 0, \tag{4.18}$$
$$C(t,x,y,+\infty) = 0. \tag{4.19}$$

Using the same procedure of MoL framework as in case of Ermak's steady state equation, the following discretized model is obtained from its time-dependent variant when $x$ derivative is approximated by the central difference

$$\begin{aligned}\frac{\partial C(t,i,j,k)}{\partial t} = &-\frac{a_x}{2\Delta x}\left[C(t,i+1,j,k) - C(t,i-1,j,k)\right] + \\ &\frac{D_y(x)}{\Delta y^2}\left[C(t,i,j+1,k) - 2C(t,i,j,k) + C(t,i,j-1,k)\right] + \\ &\frac{D_z(x)}{\Delta z^2}\left[C(t,i,j,k+1) - 2C(t,i,j,k) + C(t,i,j,k-1)\right] + \\ &\frac{W}{2\Delta z}\left[C(t,i,j,k+1) - C(t,i,j,k-1)\right].\end{aligned} \tag{4.20}$$

The appropriate boundary conditions in terms of the MoL discretization are

$$\begin{aligned}C(t,0,0,h_s) &= \frac{Q}{a_x \Delta y \Delta z} \\ C(t,0,j,k) &= 0 \quad \text{otherwise},\end{aligned} \tag{4.21}$$

$$C(t,i,+N_j,k) = 0, \tag{4.22}$$

$$C(t,i,-N_j,k) = 0, \tag{4.23}$$

$$C(t,i,j,+N_k) = 0, \tag{4.24}$$

$$C(t,i,j,0) = \frac{D_z(\infty)}{2\Delta z(v-W) + D_z(\infty)} C(t,i,j,2). \tag{4.25}$$

The evaluation of the model is presented further in section 5.1.1 and it was published in [DZ08b] (basic evaluation), [KDZM08] (includes 3D visualization) and [DZ08a] (includes error analysis).

### 4.1.2 Stability assumptions

Since numerical solution of investigated equation is designed and it has a form of ODE the stability assumptions can be approximated using von Neumann stability analysis. The consistency of the numerical solution can be assumed from the fact that partial derivatives are approximated by finite differences derived from the Taylor series expansions [26].

#### The used model stability analysis

In order to analyse the stability of the equation (4.15), the von Neumann stability analysis was made. It analyses theoretical bounds of coefficients in the discretized domain of a linear equation [25]. Let us do this procedure for a case, when the explicit Euler method is used for solving the system of ODE (4.20). This is equivalent to the well known forward time and central space method (FTCS).

Let advection coefficients, $\alpha_a$ and $\alpha_W$, and diffusion coefficients, $\beta_y$ and $\beta_z$ are defined as

$$\begin{aligned} \alpha_a &= \frac{a_x \Delta t}{2\Delta x}, \quad & \alpha_W &= \frac{W \Delta t}{2\Delta z}, \\ \beta_y &= \frac{D_y \Delta t}{(\Delta y)^2}, \quad & \beta_z &= \frac{D_z \Delta t}{(\Delta z)^2}. \end{aligned} \tag{4.26}$$

Further, lets consider wave numbers $P_x$, $P_y$ and $P_z$ in $x-$, $y-$ and $z-$directions and the appropriate phase angles that are defined as

$$\Theta = P_x \Delta x, \quad \Phi = P_y \Delta y, \quad \Psi = P_z \Delta z. \tag{4.27}$$

Then, the transformation to phase space is defined as

$$C(t, i, j, k) = U(t) e^{I(\Theta i + \Phi j + \Psi k)}, \tag{4.28}$$

where $U(t)$ is amplitude at time $t$ and $I^2 = -1$.

After substituting terms (4.26) into equation (4.20), applying transformation (4.28) and the calculations which are presented in appendix A, the following equation is obtained

$$\begin{aligned} U(t + \Delta t) =& [1 + 2\beta_y(\cos\Phi - 1) + 2\beta_z(cos\Psi - 1) \\ & - I(2\alpha_a \sin\Theta - 2\alpha_W \sin\Psi)] \cdot U(t) \\ =& G \cdot U(t). \end{aligned} \tag{4.29}$$

For a stable solution, the absolute value of amplification factor $G$ has to be bounded for all values of $\Theta$, $\Phi$, $\Psi$. Since $G$ is a complex number, then the following condition has to be satisfied for numerical stability

$$|G|^2 = [1 + 2\beta_y(\cos\Phi - 1) + 2\beta_z(\cos\Psi - 1)]^2 + (2\alpha_a \sin\Theta - 2\alpha_W \sin\Psi)^2 \leq 1. \tag{4.30}$$

After simple manipulation of inequality (4.30), we get

$$\begin{aligned} \beta_y \cos\Phi + \beta_z \cos\Psi - (\beta_y + \beta_z) + [\beta_y \cos\Phi + \beta_z \cos\Psi - (\beta_y + \beta_z)]^2 \leq \\ -(\alpha_a \sin\Theta - \alpha_W \sin\Psi)^2. \end{aligned} \tag{4.31}$$

The left side of the inequality (4.31) is a quadratic function of the form $f(x) = x^2 + x$ and its right side is always negative. From these facts the following condition must be satisfied

$$-1 \leq \beta_y \cos\Phi + \beta_z \cos\Psi - (\beta_y + \beta_z) \leq 0. \tag{4.32}$$

The second inequality of (4.32) is satisfied for every $\beta_y$ and $\beta_z$ assuming that they are non-negative – it is true according to their definitions (4.26). The first inequality, and thus the whole condition for every $\Phi$ and $\Psi$, holds for

$$\beta_y + \beta_z \leq \frac{1}{2}. \tag{4.33}$$

Next, the absolute value of the right hand side of inequality (4.31) is maximal for $\Theta = \frac{\pi}{2}$ and $\Psi = \frac{3\pi}{2}$ and it is a limiting case

$$(\beta_y + \beta_z - \beta_y \cos\Phi)(-\beta_y - \beta_z + \beta_y \cos\Phi + 1) \geq (\alpha_a + \alpha_W)^2. \tag{4.34}$$

From the fact that left hand side of inequality (4.34) corresponds to function $f(x) = -x^2 + x$, then it has a maximum of $\frac{1}{4}$ and new condition must be at least

$$\alpha_a + \alpha_W \leq \frac{1}{2}. \tag{4.35}$$

However, it is not satisfied for every case of the left side of equation (4.34). Since the condition (4.33) holds, the left side of (4.34) is minimal for $\cos\Theta = 1$ and then the final condition is

$$\alpha_a + \alpha_W \leq (1 - \beta_z)\beta_z. \tag{4.36}$$

### Evaluation

The theoretical stability conditions that are derived in previous section were evaluated in several experiments.

Figure 4.1 shows the areas where the solution is stable or not. The particular areas are marked with color of different grey level. On the left image, the advection numbers, defined in (4.26), are varying according to the sizes of $W$ and $a_x$ coefficients and thus the stability space is altered appropriately. The condition (4.35) bounds the area of $A_1$ as a stable, therefore $A_0$ is unstable. However, after applying of (4.36) the stable area is further reduced into $A_2$, in which all stability criteria are satisfied. The experiments fit to the area of $A_3$ which is part of $A_2$, thus the theoretical requirements are correct.

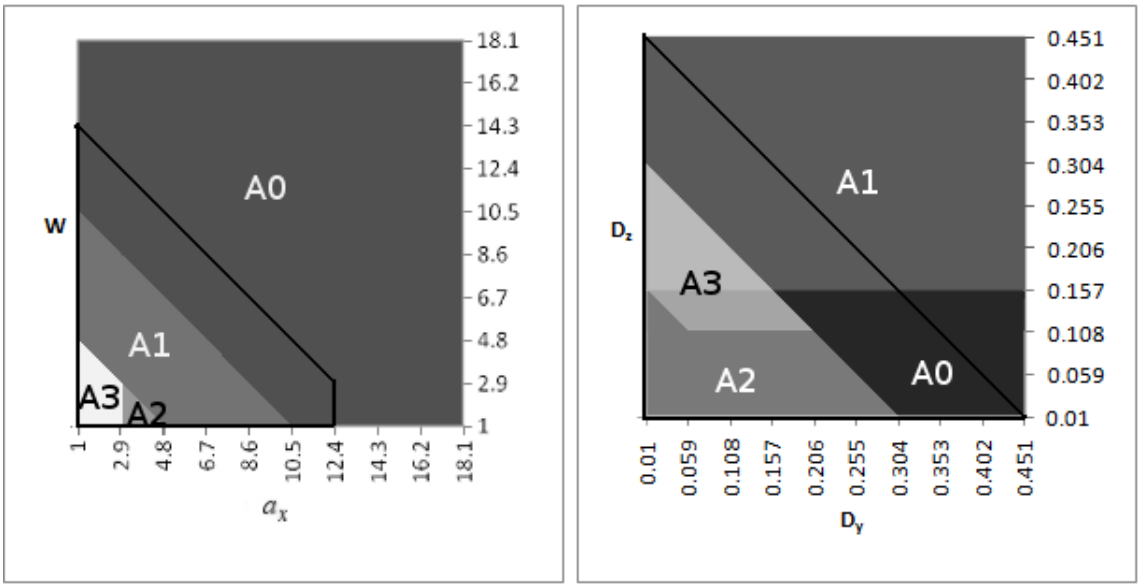



Figure 4.1: The areas of stability for advection (left) and diffusion (right).

On the right side of figure 4.1, various diffusion numbers, defined in (4.26), are investigated. Here, the coefficients in the area of $A_1$ satisfy the condition (4.36) and the coefficients of $A_2$ satisfy (4.33). The intersection of these areas $A_1$ and $A_2$ has the brightest color and is a little smaller than the area of $A_3$, which was obtained by the experiments. Again it

corresponds with each other up to some numerical errors. In both graphs in figure 4.1 the areas bordered with black line belong to stable space of solutions where the $4^{th}$-order Runge-Kutta method was used. It is obvious that the area of stability is much larger than in case of FTCS method.

The presented results were published in [DZ09].

### 4.1.3 Parallel design

A fundamental challenge for probably every branch of research in computer science and engineering area has been the ongoing quest for a higher performance. From a historical point of view, computer software has been mainly designed with a serial style of processing in mind. Parallel computing on the other hand relies on multiple processing elements at once to solve a given task. The problem is, therefore, broken into parts which are independent so that each processing element can handle its portion of the algorithm alongside the others. In that sense, specialized computational architectures have been exposed to a closer attention. These include GPUs, IBM Cell platform or FPGA-based devices. However, one should take into account that the significant performance benefits can be unleashed only for a specific class of problems. In this section the concrete design of the MoL numerical solution for GPU platform will be described.

#### Related work

During the past few years, various papers dealing with the exploitation of GPUs for general-purpose computing tasks have emerged. For example, in a paper of Brandvik et al. [6], the authors have carried out numerous experiments with Euler solver implementation with BrookGPU and CUDA platforms. They reached the 29× speed-up in 2D with BrookGPU and 19× speed-up factor in 3D with CUDA.

Micikevicius [37] studied the ways, how to implement the finite difference approaches on single and multiple GPUs. He was able to reach an order of speed-up against high end CPU and linear scaling communication overhead when using multiple GPUs. In the work by Datta et al. [15], an optimal stencil computation kernel is implemented under CUDA. The performance evaluation is given through the number of points calculated per second. Several architectures were compared and the GPU platform was found to be the most efficient one.

Almost the same problem of air pollution and the possible way of computational acceleration is studied in work from Molnar et al. [39]. The model in their work was based on stochastic model rather than ADE approach used in this thesis. Nevertheless, they were able to reach 80-120 times faster computational time on single GPU than on CPU.

In the work by Tolke et al. [55], the lattice Boltzmann kernel is implemented with CUDA and the performance increase of approximately 100× has been successfully reached. Other application of CUDA usage in computational fluid dynamics is presented by Molemaker et al. [38]. It is used to calculate simulation model progress of low viscous fluid. Achieved speedup is about 55×.

The works presented here were a big motivation to implement and test the concrete parallel version of the numerical model based on MoL framework described in this work.

### CUDA/OpenCL architectures

The GPUs are especially well suited to address the problems that exhibit a data-parallel nature with high arithmetic intensity. Until recently, the communication link between a program running on the CPU and the graphics hardware had to be established by means of using a graphics API such as OpenGL. To perform computations on graphics hardware, the programmer was obliged to use graphic primitives and store the data in texture maps. Luckily, the advent of transparent and flexible programming frameworks, like CUDA [12] in 2007 or OpenCL [20] at the very end of 2008, which provide an interface between GPU and CPU host machine, enabled the majority of the aforementioned drawbacks to be mitigated. Moreover, the introduction of OpenCL framework has enabled the possibility to write a single piece of code that can be further launched on various compatible architectures.

CUDA nad OpenCL frameworks have to be seen as a fusion of SW and HW parts. Lets first consider and example of GeForce GTX2xx series cards which were further used in our tests. It offers 240 stream processing elements organized into a collection of 30 identical multiprocessors. Each multiprocessor has its own shared memory, which is common to all the 8 processors inside. It also has a set of 32-bit registers, texture, and constant memory caches. At any given cycle, each element in the multiprocessor executes the same instruction on different data, which makes each a SIMD processor. Communication between multiprocessors is carried out through the device memory, which is available to all the processors of the multiprocessors.

From the SW point of view, a target application for CUDA/OpenCL is based on a collection of threads running in parallel. The computation is distributed in a grid of thread blocks (work-groups in case of OpenCL). All blocks contain the same number of threads that execute a program, known as kernel, on a device. Kernels are called work-items in case of OpenCL terminology. It is possible to use a 1D, 2D, or 3D index space to invoke and keep a hand on the kernel.

Several different types of memory are available in the frameworks. Global memory can be accessed by every work-item on the compute device, which mostly offers the slowest access speed and largest size. The purpose of global memory/constant memory cache is to improve the necessary latency associated with data transfers. It is readable only for most of the time. Local memory, which is available to all work-items in the same work-group, offers much faster speed than global memories. However, its size is very limited. Private memory of a work-item, which is only accessible by itself, has the lowest latency but most limited storage space.

The efficiency of a kernel can be significantly improved by taking an advantage of parallel access to shared memory and by avoiding bank conflicts. The performance of iterative or multi-phase algorithms can be improved if all the computations can be performed in the GPU, so that step 3 bellow can be run several times without the need to exchange the data between device and host. A typical algorithm execution flow for the frameworks consists of the following stages:

1. Allocate data on the device.
2. Transfer data from the host to the device.
3. Proceed with the execution of kernel(s). The result is stored in device (local) memory.
4. Retrieve data from device and transfer them to the host environment.

#### Outline of the solution

The selected numerical model of our solution divides the space into a finite set of discrete points where the concentration level is calculated for each of them. With regards to the fact that $z$ variable remains continuous and the PDE (4.9) assumes steady state form (the concentrations remain constant during time), the number of equations is simply quantified as $(N_j) \times (N_k)$. In case of time-dependent variant of PDE (4.20) the number of equations is obtained as $(N_i) \times (N_j) \times (N_k)$. Thus the amount of memory used for calculation on graphics hardware is clearly a multiple of this number. Of course, some auxiliary variables have also to be taken into account.

The gist of numerical solution includes three main parts. In every integration step, the derivative (4.9) for steady-state case or the derivative (4.20) for time-dependent case are calculated. The derivative value has to be determined four times during the integration phase due to the requirement imposed by the principle of the fourth-order Runge-Kutta method. The last stage of the algorithm is responsible for a correct handling of boundary conditions. First two stages are done in a sequential manner with one thread assigned for each equation inside the specified discrete point space. Subsequently, the boundary values are calculated for boundary points only (see equations 4.14, 4.13, 4.25, 4.24).

Both arrangements of work-groups within the index space and configuration of kernels inside each work-group have an impact on the resulting performance. The usage of one-dimensional indexing of kernel has an advantage of reducing floating point operation to obtain the kernel location. Thus, it is worth to use it instead of 2D or 3D indexing (the performed evaluation also confirmed that). On the other hand, one-dimensional index reduces the number of addressable kernels. The dimensions of the index space and work-groups, respectively, are $[65535, 65535, 1]$ and $[512, 512, 64]$. The highest number of threads in each work-group is therefore $2^{32} \times 2^9 = 2^{41}$ (maximum block size is 512). In case of one-dimensional index, the maximum number of threads is reduced to $2^{16} \times 2^9 = 2^{25}$, which has to be taken into account during implementation.

The detailed evaluation is presented in section 5.1.2. The detailed results were also published in [SDZK09] (CUDA), [SDZD09] (CUDA, time-dependent variant) and [SDZD10] (CUDA, OpenCL and comparison).

### 4.1.4 Wortmann's advection-diffusion model

The second model used for MoL evaluation is the model presented by Wortmann et al. [60]. It is the relatively simple steady-state model depicted by equation (4.37). More specific, it is the PDE in two dimensions

$$a(z)\frac{\partial C}{\partial x} = D_z\frac{\partial^2 C}{\partial z^2} + \left(\frac{\partial D_z}{\partial z}\right)\frac{\partial C}{\partial z}, \tag{4.37}$$

where $C$ (kg m$^{-3}$) is concentration, $a(z)$ (m s$^{-1}$) is wind velocity, $D_z$ (m$^2$ s$^{-1}$) is vertical diffusion coefficient.

The needed boundary conditions are defined as follows

$$D_z\frac{\partial C}{\partial z} = 0 \quad \text{for } z \in \{0, H\}, \tag{4.38}$$

$$a(z)C(0, z) = Q\delta(z - h_s) \quad \text{for } x = 0, \tag{4.39}$$

where $H$ (m) is domain height, $Q$ (kg s$^{-1}$) is point source emission rate and $h_s$ (m) is point source height.

### Analytical solution

In analytical solution presented by Wortmann et al. [60] the procedure to calculate the pollutant concentration is based on the application of a general integral transform technique (GITT) and it is fairly complicated. The expression, by which the result is obtained, is of the form

$$C(x,z) = \sum_{i=0}^{\infty} \frac{\overline{c_i}(x)\Psi_i(z)}{N_i^{1/2}}. \tag{4.40}$$

Variable $\overline{c_i}(x)$ is a vector and it is calculated by a procedure that involves a great deal of linear algebra operations, such as matrix inversion, multiplication of matrices, etc. The other two variables $\Psi_i(z)$ and $N_i^{1/2}$ represent simple algebraic expression and integration (numerical probably), respectively. Thus the calculation requires the deployment of a huge computation power in order to get the concentration amount for a given point. Moreover, good approximation requires many terms of a series (4.40) and therefore, the matrix size involved in the calculation process is getting larger with each additional term. Nevertheless, the analytical solution can be used as a verification of a numerical one for the special cases.

### MoL solution

The numerical solution obtained by the method of lines has very simple form. The aim of the process is to construct a new equation that approximates the original one. To do that the spatial derivatives are replaced by final differences but one of them is retained in continuous state. The form of ordinary differential equation is then acquired and it can be solved by any numerical integration method such as Euler, Runge-Kutta and others in implicit or explicit form.

In our case the equation after transformation is as follows

$$\begin{aligned}\frac{\partial C}{\partial x} =& \frac{D_z}{a(z)} \cdot \frac{C(x,z+1) - 2C(x,z) + C(x,z-1)}{\Delta z^2} + \\ & \frac{1}{a(z)} \cdot \frac{\partial D_z}{\Delta z} \cdot \frac{C(x,z+1) - C(x,z-1)}{2\Delta z}.\end{aligned} \tag{4.41}$$

Here, $\Delta z$ is step size in $z$ direction. The appropriate boundary conditions using central differences result in

$$C(x,0) = C(x,2), \tag{4.42}$$

$$C(x,h) = C(x,h-2), \tag{4.43}$$

$$C(0,h_s) = \frac{Q}{a(z)\Delta z}. \tag{4.44}$$

The evaluation of presented model and comparison with various analytical solutions is presented in section 5.1.3. The results were also published in [DZS09a].

## 4.2 Form of ELLAM

The second method which this thesis is mainly focused on is ELLAM whose general concepts are described in section 3.3.4.

The formulation of ELLAM framework was originated around 1990 by the authors Herrera and Ewing in a paper that appeared in Advances in Water Resources [10] where its superior performance was shown in one-dimensional ADE case with constant coefficients. Since that time the method was applied to many other more complex problems in 2 or 3 dimensions (see for instance [3], [61], [62], [36], [34]). In all of these cases, ELLAM performs well because it combines a Lagrangian approach for the advective terms with appropriate approximations, consistent with the Lagrangian framework, for other terms in the equations.

In this section, the description of concrete form which is used in the thesis is described in more detail. Moreover, the techniques to avoid oscillations and to improve the method accuracy are described in appropriate subsection.

### 4.2.1 Basic concepts

The form of ELLAM framework used in this thesis is formulated for two-dimensional space domain ($\Omega$). Leading by the presented contaminant models, it is designed for the advection-diffusion equation and for the simplicity it uses a rectangular grid. The concrete implementation is inspired by the work of Liu [33] where the space discretization is based on the finite element method. The governing equation in this case is defined as

$$\frac{\partial C}{\partial t} - \nabla \left( \vec{a} C - \vec{D} \nabla C \right) = S\left(\vec{x}, t\right), \quad \vec{x} \in \Re^2, t > 0, \tag{4.45}$$

where $S$ (kg m$^{-3}$ s$^{-1}$) is function of the source of the pollution, $C$ (kg m$^{-3}$) is concentration, $\vec{a}$ (m s$^{-1}$) is velocity field, $\vec{D}$ (m$^2$ s$^{-1}$) is diffusion vector and $t$ (s) is time. In the following equations, time $t$ at step $n$ will be denoted as $t_n$ and previous time as $t_{n-1} = t_n - \Delta t$, where $\Delta t$ is step size. The resulting weak formulation for the specified time $t_n$ after multiplication by test function $g(\vec{x}, t)$ and applying of Green's formula is

$$\begin{aligned} \int_\Omega \left(Cg\right)\left(\vec{x}, t_n\right) d\vec{x} + \int_{J_n} \int_\Omega \left( \vec{D} \nabla C \right) \nabla g\left(\vec{x}, t\right) d\vec{x} dt + \\ \int_{\Gamma_n} \left( \vec{a} C - \vec{D} \nabla C \right) \cdot \vec{n} g(\vec{y}, t) d\vec{y} dt = \\ \int_\Omega \left(Cg\right)\left(\vec{x}, t_{n-1}\right) d\vec{x} + \int_{\Sigma_n} S\left(\vec{x}, t\right) g\left(\vec{x}, t\right) d\vec{x} dt, \end{aligned} \tag{4.46}$$

where $\vec{n}$ is normal outward unit vector from the element $dydt$, $J_n$ is time domain, $\Sigma_n = \Omega \times J_n$ is space-time domain, $\Gamma_n = \partial\Omega \times J_n$ is boundary domain, $d\vec{y}dt \in \Gamma_n$ and $g\left(\vec{x}, t_{n-1}\right) = \lim_{t \to t_{n-1}} g\left(\vec{x}, t\right)$. The second integral on the left hand side of the equation (4.46) is a diffusion term, the third integral is a boundary term and the second integral on the right hand side is a source term.

To evaluate the equation (4.46), the following procedure is done. The test function $g$ was chosen as in one-dimensional case (3.92) extended to two dimensions, which results to the following form

$$g^n_{i,j}(\vec{x},t) = \begin{cases} \left(\frac{x-x_{i-1}}{\Delta x} + a_x \frac{t_n - t}{\Delta x}\right) \cdot \left(\frac{y-y_{j-1}}{\Delta y} + a_y \frac{t_n - t}{\Delta y}\right) \\ \quad x^*_{i-1} \le x \le x^*_i, y^*_{j-1} \le y \le y^*_j, t_{n-1} < t < t_n \\ \left(\frac{x_{i+1}-x}{\Delta x} + a_x \frac{t_n - t}{\Delta x}\right) \cdot \left(\frac{y_{i+1}-y}{\Delta y} + a_y \frac{t_n - t}{\Delta y}\right) \\ \quad x^*_i \le x \le x^*_{i+1}, y^*_j \le y \le y^*_{j+1}, t_{n-1} < t < t_n \\ 0 \\ \quad \text{for other } x, t. \end{cases} \tag{4.47}$$

where $(x^*_k, y^*_l)$ $(k \in \{i-1, i, i+1\}, l \in \{j-1, j, j+1\})$ are points at time $t_{n-1}$ corresponding to points $(x_k, y_l)$ at time $t_n$ along the characteristic curve.

The terms with time integration are discretized using backward Euler method. The source term of the equation is approximated as

$$\begin{aligned} \int_{\Sigma_n} S\left(\vec{x},t\right) g\left(\vec{x},t\right) d\vec{x}dt = \\ \int_{\Omega} \Delta t^I(\vec{x},t_n) S(\vec{x},t_n) g(\vec{x},t_n) d\vec{x} + \\ \int_{\Gamma^O_n} \Delta t^O(\vec{y},t) S(\vec{y},t) g(\vec{y},t) (\vec{a}\cdot\vec{n}) d\vec{y}dt + E(S,g), \end{aligned} \tag{4.48}$$

where $\Delta t^I(\vec{x},t_n) = t_n - t_{n-1}$, $\Delta t^O(\vec{y},t) = t - t_{n-1}$ and $E(S,g)$ is error of the approximation. Similarly, the diffusion term can be approximated as follows

$$\begin{aligned} \int_{J_n}\int_{\Omega} \left(\vec{D}\nabla C\right) \nabla g(\vec{x},t) d\vec{x}dt = \\ \int_{\Omega} \Delta t^I(\vec{x},t_n) \left(\vec{D}\nabla C\right) \nabla g(\vec{x},t_n) d\vec{x} + \\ \int_{\Gamma^O_n} \Delta t^O(\vec{y},t) \left(\vec{D}\nabla C\right) \nabla g(\vec{y},t) (\vec{a}\cdot\vec{n}) d\vec{y}dt + E(\vec{D},C,g), \end{aligned} \tag{4.49}$$

where $E(\vec{D},C,g)$ is approximation error.

Assuming no boundary terms in the implementation, the corresponding integrals and error terms can be dropped and therefore final equation for one element will have the form

$$\begin{aligned} \int_{\Omega} \left(Cg\right)\left(\vec{x},t\right) d\vec{x} + \int_{\Omega} \Delta t^I(\vec{x},t_n) \left(\vec{D}\nabla C\right) \nabla g(\vec{x},t_n) d\vec{x} = \\ \int_{\Omega} \left(Cg\right)\left(\vec{x},t_{n-1}\right) d\vec{x} + \int_{\Omega} \Delta t^I(\vec{x},t_n) g(\vec{x},t_n) g(\vec{x},t_n) d\vec{x}dt. \end{aligned} \tag{4.50}$$

The integrals in equation (4.50) are evaluated by numerical integration using Gaussian quadrature with appropriate integration points. It remains to evaluate the equation $g\left(\vec{x},t_{n-1}\right) = \lim_{t\to t_{n-1}} g\left(\vec{x},t\right)$. This problem leads to the solution of the ordinary differential equation back in time. The common integration methods such as Euler method defined by equation (3.14) can be used. The $4^{th}$-order Runge-Kutta method is used in the current

implementation. It is a trade-off between speed and accuracy and it behaved very well in the cases of the performed experiments.

The last thing to explain is the space discretization. The rectangular grid of points and the standard FEM process is used. The equation (4.50) has to be solved on the whole domain, therefore the elements, on which the approximation of the unknown function $C$ is defined, have to be assembled together. This leads to the system of algebraic equations that has to be solved at each time step.

As a summary, the procedure of the ELLAM method can be expressed as:

- Initialization. Assembly of elements into global matrix.
- Loop. In each iteration step, till $t < t_{end}$ holds, do:
  1. assembly source term to right-hand side of set of equations.
  2. assembly the global right-hand side for the old mass (characteristic tracking to time $t_{n-1}$).
  3. optionally, modify global system with boundary condition term.
  4. solve global system of equations for $t = t_n$.

The evaluation of the described form of ELLAM is presented in section 5.2. The results were also published in [DZ10].

### 4.2.2 Oscillation reduction

As it was shown in simple one-dimensional example in section 3.3.4 ELLAM often suffers from excessive numerical oscillations around the exact solution. The standard approach for this problem is to use *mass lumping* which fixes the oscillations but increases numerical diffusion [3]. In work of Russel and Binning [45] the *selective mass lumping* procedure is designed for 1D case which significantly reduces the arisen numerical diffusion.

Next to these, there actually exist two works that particularly deal with the problem. In the first one [40], the authors use lumping technique and a post-processing procedure, a change of the matrix representing the final system of equations in a way to preserve monotonicity properties of the solution. They showed the very good results of global overshoot reduction on problems in two dimensions on structured and unstructered grids.

The second work which deals with the problem of excessive numerical diffusion added when using mass lumping techniques is the one by Younes at al. [63]. Here, the authors use mass lumping and keep the same characteristics during the entire simulation. At the end of each time step the diffusion part is added using the interpolation technique (only the diffusion part is interpolated). The excellent properties of the technique were demonstrated in one-dimensional test cases.

By inspiration of Walcek method and based on experience with the ELLAM simple experiments few techniques to reduce oscillations and/or improve ELLAM method accuracy were designed in this work. They are described in the following subsections.

#### Simple flux limiting

The idea of the first method is to use the simple flux limiting approach similar to one in Walcek scheme where the simple rules presented in section 3.2.3 are applied. If Courant number is less than 1 the amount of transferred mass to receiving cell cannot be bigger or

less than the maximum or minimum concentrations in both donor and receiver cells. If the violation of these restrictions is detected the amount of concentration changed accordingly.

With such procedure, there is no guarantee that total amount of mass will be preserved, a certain amount of concentration will be dropped off. The question is what to do with this residual. During simple experiments of advection equation in one dimension using sine- and box-shaped concentration profiles, it was observed that an approach of homogeneous distribution of residual gives the good results. Actually, the distribution is done selectively to the cell with the reasonable amount of concentration (at least 0.01% of peak concentration) or a big change of concentration with respect to previous time step.

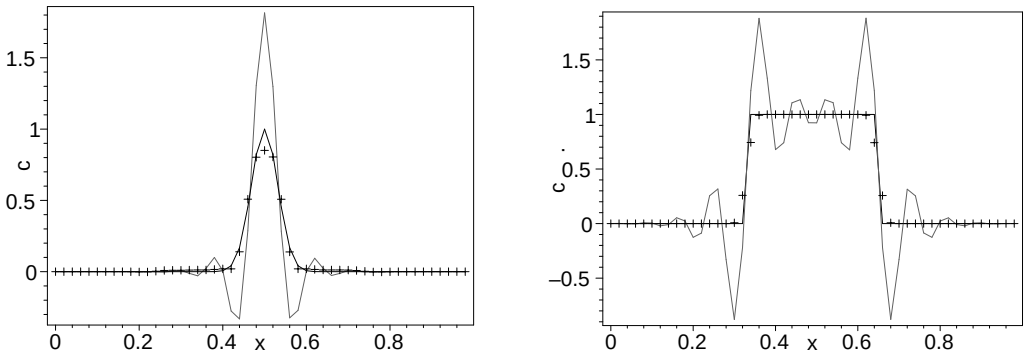



Figure 4.2: Test result of ELLAM scheme with simple limiter for sine- (left) and box-shape (right) initial profiles. Black solid line is the exact solution, grey solid line is the original scheme and cross are points of the scheme with the limiter.

The results for one-dimensional experiments of the described limiting method are shown in figure 4.2. It could be seen there that the excessive numerical oscillations are avoided with the minimum additional diffusion (it is shown in the same figure when compared with the exact solution).

### Using artificial diffusion

In general, all flux limiters add kind of artificial diffusion to the method in order to avoid oscillations - the simple example is presented in figure 3.3. Therefore, why not to add selectively the diffusion to the currently running simulation when needed. Here is important to define proper criteria and proper amount of artificial diffusion in order to not add too much or not too less. The big amount of diffusion leads to inaccurate shape preserving in case of pure advection problems. On the other hand, too less diffusion has a consequence of still existing oscillations and thus inaccuracy or even instability. The similar approach was used in [45] to use mass lumping selectively to avoid large numerical diffusion.

The criteria of the amount of diffusion added used here is based on a size of oscillations detected in current time step during a simulation. The question is how to determine the oscillation size. Because it is physically impossible to reach negative concentrations, these are very good indications of oscillations. Secondly, when model describes pure advection or advection-diffusion process without source or sink terms the criteria of global maximum can be used. In this case it is physically impossible to get global concentration maximum higher than at each step of simulation. The criteria can be in general stated in the following form

$$oscillations \begin{cases} \text{if } C^i_{t_n} < -C^{max}_{t_n}\mu \\ \text{if } C^i_{t_n} > C^{max}_{t_n}(1+\mu), \end{cases} \tag{4.51}$$

where $i$ is the cell index, $t_n$ denotes time step $n$ and $\mu \geq 0$ is the defined oscillation detection ratio. The size of the oscillations is important to final decision if the artificial diffusion has to be increased or not. In order to obtain a relative oscillation size (a ratio) $o_r$, a simple equation can be used

$$o_r = \max_i(o_i/c^{max}_{t_n}), \tag{4.52}$$

where $i$ is the cell index. The actual algorithm to adjust the diffusion in current time step $t_n$ is then done through the iteration process where the current diffusion coefficient is adjusted using bisection algorithm to find the optimal value.

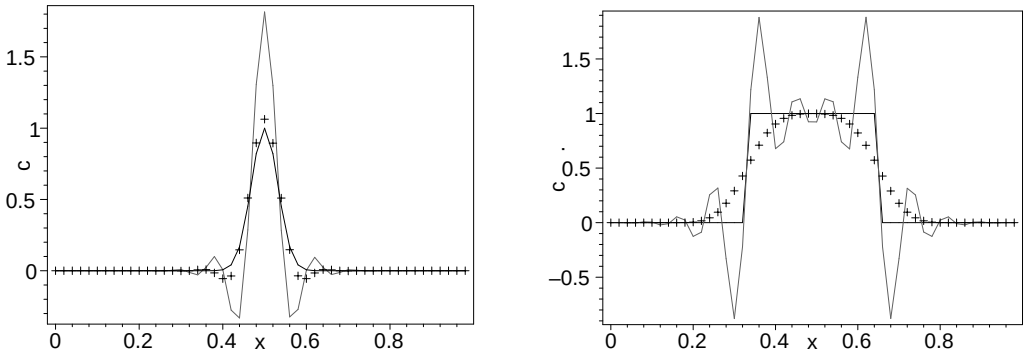



Figure 4.3: Test result of ELLAM scheme with selective artificial diffusion added for sine- (left) and box-shape (right) initial profiles. Black solid line is the exact solution, grey solid line is the original scheme and cross points are of the scheme with selective diffusion.

The results of the presented technique are shown in figure 4.3. It can be seen there that the sine shape of concentration profile is quite well preserved, however, the case of sharp gradients (the case of box-shaped profile) indicates that big diffusion had to be used in order to avoid oscillations. The question here is whether the amount of diffusion was not set too high. In more advanced experiments it was shown that it could be better in many cases to keep smaller oscillations to preserve proper concentration shape. The actual results are shown in section 5.2.

### Time-step adaptation

During implementation and testing of the form of ELLAM used in this thesis it was observed that bigger steps lead to significantly smaller oscillations and thus better accuracy - an example is the usage of quantiple step size in section 3.3.4. This is with accordance of the results and derivations in [45], [40] and [63]. Therefore, the idea to gain better accuracy and less oscillations is to use as big step as possible during simulation - dynamic step sizing.

The criteria when to use bigger time steps is very similar to the case of oscillation avoidance using artificial diffusion. When oscillations are detected using equation (4.51) and the oscillation ratio calculated by (4.52) is lower than in previous time step the step size is doubled. The procedure continuous till the condition holds. When the increase of step size leads to bigger oscillations than before the previous time step size is used.

It is very important to keep in mind that characteristic tracking has to remain accurate when step size is modified. Thus when it is increased the number of micro time steps of method for characteristic tracking has to be accordingly adjusted. One can use the technique similar to adaptive time-stepping as in case of MoL experiments with real coefficient models (see section 5.1.3). In current solution, the size of micro steps is deduced from the factor of time step increase. The actual micro-step size $dt_n$ was empirically determined as

$$dt_n = \begin{cases} 1/(N_p * 1.35), & dt_n > dt_p \\ 1/(N_p/1.35), & \text{otherwise}, \end{cases} \tag{4.53}$$

where $dt_p$ is micro step size from previous iteration and $N_p = 1/dt_p$ is number of previous micro steps.

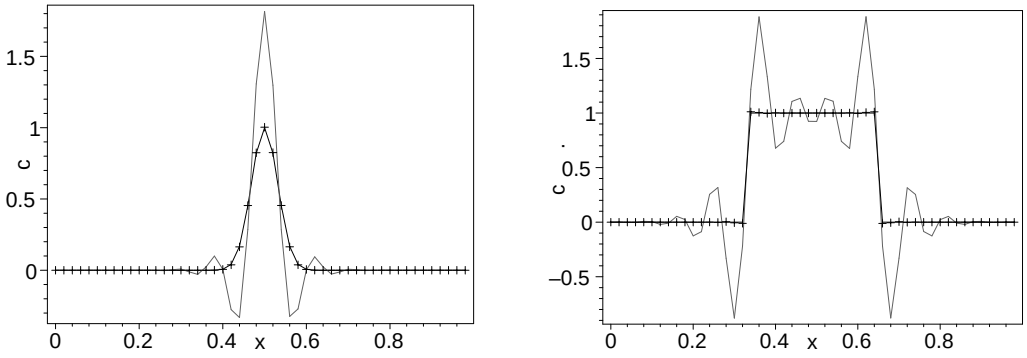



Figure 4.4: Test result of ELLAM scheme with dynamic time-step adaptation for sine- (left) and box-shape (right) initial profiles. Black solid line is the exact solution, grey solid line is the original scheme and cross points are of the scheme with adaptive time-stepping.

The results of presented time-stepping method for simple 1D experiment are shown in figure 4.4. It is worth to do time-adaptation in case of ELLAM which is shown by this example. The more advanced tests in 2D are presented in dedicated section 5.2 which confirm this conclusion.

### Summary

As a summary of the section, a set of experiments was done in order to confirm the correctness of the three approaches. The tests were done for different time step sizes from 0.005 s to 1 s for diffused and time-adaptation variants, and from 0.005 s to 0.01 s for the flux limiter variant in order to not violate the CFL condition.

The results of the experiments are shown in table 4.1 for sine-shaped wind profile and in table 4.2 for box-shaped wind profile.

It could be clearly seen that in terms of accuracy the adjustments of the individual limiter and diffusive methods slightly improve original ELLAM in average cases, however, they bring noticeable improvements for worst case. The time-adaptation approach has significantly best results than others including Walcek method.

The advantage of the designed improvements is their generality for any form of ELLAM using structured grids. Diffusion and time-adaptive variants can be from their principle directly used also for unstructured grids. Moreover, the approaches are not dependent on

| Scheme | $L_1$-norm error | $L_2$-norm error | $L_\infty$-norm error |
|---|---|---|---|
| Walcek | $0.736 \cdot 10^{-2}$ | $0.361 \cdot 10^{-3}$ | $0.792 \cdot 10^{-1}$ |
| ELLAM average | $0.174 \cdot 10^{-1}$ | $0.300 \cdot 10^{-2}$ | 0.186 |
| ELLAM maximum | $0.729 \cdot 10^{-1}$ | $0.298 \cdot 10^{-1}$ | 0.718 |
| ELLAM average (diffusion) | $0.138 \cdot 10^{-1}$ | $0.175 \cdot 10^{-2}$ | 0.150 |
| ELLAM maximum (diffusion) | $0.297 \cdot 10^{-1}$ | $0.544 \cdot 10^{-2}$ | 0.331 |
| ELLAM average (limited) | $0.341 \cdot 10^{-1}$ | $0.665 \cdot 10^{-2}$ | 0.379 |
| ELLAM maximum (limited) | $0.440 \cdot 10^{-1}$ | $0.987 \cdot 10^{-2}$ | 0.461 |
| ELLAM average (time-adapted) | $0.568 \cdot 10^{-2}$ | $0.464 \cdot 10^{-3}$ | $0.682 \cdot 10^{-1}$ |
| ELLAM maximum (time-adapted) | $0.291 \cdot 10^{-1}$ | $0.522 \cdot 10^{-2}$ | 0.324 |

Table 4.1: $L_n$-norm error measures of performed one-dimensional advection experiments of Walcek and ELLAM methods with various oscillation avoidance techniques in case of sine-shape initial profile.

| Scheme | $L_1$-norm error | $L_2$-norm error | $L_\infty$-norm error |
|---|---|---|---|
| Walcek | $0.122 \cdot 10^{-1}$ | $0.188 \cdot 10^{-2}$ | 0.153 |
| ELLAM average | $0.485 \cdot 10^{-1}$ | $0.122 \cdot 10^{-1}$ | 0.334 |
| ELLAM maximum | 0.157 | $0.676 \cdot 10^{-1}$ | 0.769 |
| ELLAM average (diffusion) | $0.823 \cdot 10^{-1}$ | $0.239 \cdot 10^{-1}$ | 0.401 |
| ELLAM maximum (diffusion) | 0.130 | $0.382 \cdot 10^{-1}$ | 0.452 |
| ELLAM average (limited) | $0.474 \cdot 10^{-1}$ | $0.137 \cdot 10^{-1}$ | 0.381 |
| ELLAM maximum (limited) | $0.601 \cdot 10^{-1}$ | $0.177 \cdot 10^{-1}$ | 0.433 |
| ELLAM average (time-adapted) | $0.188 \cdot 10^{-2}$ | $0.152 \cdot 10^{-3}$ | $0.123 \cdot 10^{-1}$ |
| ELLAM maximum (time-adapted) | $0.224 \cdot 10^{-1}$ | $0.215 \cdot 10^{-2}$ | 0.138 |

Table 4.2: $L_n$-norm error measures of performed one-dimensional advection experiments of Walcek and ELLAM methods with various oscillation avoidance techniques in case of box-shape initial profile.

each other and can be combined which is shown in more advanced experiments in section 5.2. Some of the adaptive techniques were also published in [DZ11] and [DZ12].

# Chapter 5

# Experiments

The designed methods described in chapter 4 were put under extensive testing in various scenarios. The goal was first to verify the models against known analytical solution and secondly to use the real wind/diffusion models in order to prove whether the models are suitable for practical calculations.

The models whose solution is based on method of lines framework were tested in artificial conditions using the known analytical solution, then they were exposed to real models that contain wind and vertical diffusion models and finally a parallel version of the model in three-dimensional space was designed and tested for CUDA and OpenCL platforms.

Due to the complexity of ELLAM framework based models only two dimensional variants were implemented and tested in this work. First part of tests is dedicated to pure advection cases with rotating and divergent winds where the results of the method were compared with state of the art Walcek method described in section 3.3.3. Next set of experiments was done also with real advection-diffusion cases where the real wind models were used.

## 5.1 Method of lines

This section describes the experiments done with the designed numerical models that are defined in section 4.1. First, the simple artificial tests using discretized Ermak's model are presented followed by the experiments of parallel version of the same model and its time-dependent variant. The last part is dedicated to experiments with real models and their evaluation.

### 5.1.1 Artificial tests - model validation

The simple cases of experiments were firstly done to verify the simple steady state numerical model described by equation (4.9).

The conditions of experiments were the following. The model supposes one point source that has constant strength. It means that the amount of pollutant is constant during time. The wind flows along $x$ axis with constant speed and the ground is flat everywhere. All diffusion coefficients were constant in each space point during time to be possible to compare to known analytical solution found by Ermak [18].

The experiment has been done with the following coefficient settings which meet common atmospheric conditions. The diffusion coefficients were set as $D_y = 0.23\ \mathrm{m}^2\ \mathrm{s}^{-1}$, $D_z = 0.23\ \mathrm{m}^2\ \mathrm{s}^{-1}$ which is the parameter of ammonia, other coefficients has been set as

| Scheme | $L_1$-norm error | $L_2$-norm error | $L_\infty$-norm error |
|---|---|---|---|
| MoL | $2.979 \cdot 10^{-5}$ | $3.952 \cdot 10^{-6}$ | 2.396 |
| MoL normalized | $1.490 \cdot 10^{-6}$ | $9.879 \cdot 10^{-7}$ | 0.120 |
| MoL (time-dep.) | $1.443 \cdot 10^{-3}$ | $3.262 \cdot 10^{-5}$ | 16.382 |
| MoL (time-dep.) normalized | $7.216 \cdot 10^{-5}$ | $8.156 \cdot 10^{-7}$ | 0.819 |
| MoL (time-dep., upwind) | $3.464 \cdot 10^{-4}$ | $2.503 \cdot 10^{-5}$ | 2.197 |
| MoL (time-dep., upwind) normalized | $1.732 \cdot 10^{-5}$ | $6.257 \cdot 10^{-8}$ | 0.110 |

Table 5.1: $L_n$-norm error measures of performed artificial experiments.

$v = 2$ m s$^{-1}$, $W = 3$ m s$^{-1}$, $a_x = 2$ m s$^{-1}$, $Q = 0.1$ kg s$^{-1}$ and $h_s = 1.5$ m. The space discretization, chosen to satisfy the method stability, was set as $N_i = 600$, $N_j = 50$, $N_k = 50$, $\Delta x = 0.005$ m, $\Delta y = 0.05$ m and $\Delta z = 0.05$ m. Therefore, in this case the assumed space 3 m $\times$ 2.5 m $\times$ 2.5 m was discretized into 3090600 points in which the equations were calculated.

The $l_n$-norm errors of the performed experiment are shown in table 5.1. In the first row, the errors with direct output of methods are presented and it can be seen that there is quite large $l_\infty$-norm error when compared to strength of the source. However, this is because relative errors are measured there. The case where concentration values are normalized according to maximal value in analytical solution are shown in the second row of the table which shows that maximum absolute error was about 12%. Figure 5.1 shows the maximal errors in each $yz$ slice. It is obvious that maximum error is near the source and rapidly falls down with increasing distance.

The experiments with time-dependent variant of the model described by equation (4.20) were also performed. The conditions of the experiment were defined as before including the steady source and thus the solution remained the same. The time step of the solution was chosen to meet stability criteria according to stability analysis and was set to 0.002 s. The number of points were reduced along $x$ axis to 60 since the integration is done over time $t$. Because of steady continuous source the ending criteria was chosen such that the rate of change of concentration is minimal (stable concentration was achieved). The error measurements of the solution are again shown in table 5.1 in the third and fourth rows. It can be seen that the model has lower accuracy which is caused by central difference used to approximate $x$ derivative. The results of upwind scheme are presented in the fifth and sixth rows and they correspond to the steady state case.

### 5.1.2 Parallel versions - CUDA/OpenCL

When the model was verified with simple experiments as shown in section 5.1.1 the model parallelization and its evaluation followed. The motivation here is to improve the model computational speed in order to use more equations, and thus better accuracy within the similar or even smaller computational time.

The experiments were done on a bit older CPU and GPU platforms, however, the results still show great potential to use GPU for general purpose computation. It has to be noted that all computations were done using 32 bit floating point arithmetic [1] due to the lack of support of bigger precision on tested GPUs.

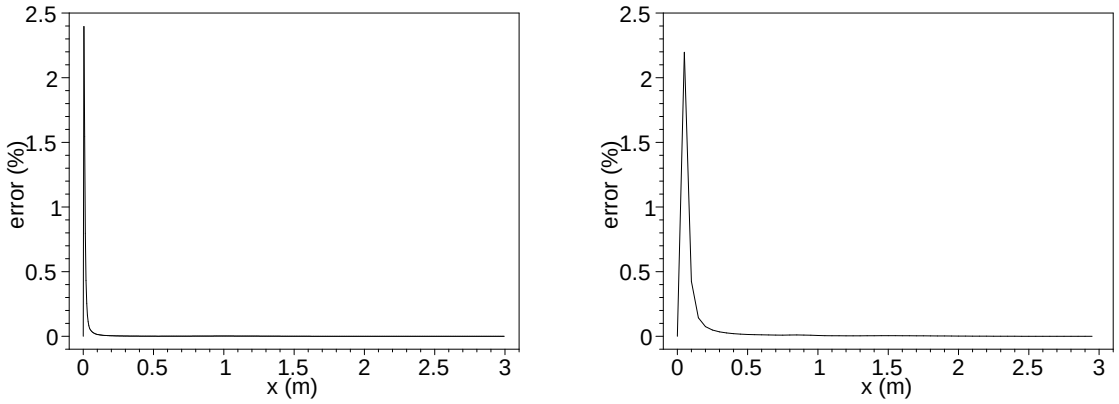



Figure 5.1: Absolute error in all $yz$ slices as a function of downwind distance for steady-state (left) and time-dependent upwind (right) experiments.

| Platform | Block size | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 4 | 8 | 16 | 32 | 64 | 128 | 256 |
| 9600M GT | 0.44 | 1.29 | 3.20 | 6.99 | 5.96 | 16.38 | 17.75 | 17.73 | 17.36 |
| 8800 Ultra | 2.31 | 6.35 | 14.94 | 28.81 | 31.46 | 56.12 | 66.08 | 57.91 | 61.18 |
| GTX280 | 4.46 | 12.29 | 26.44 | 45.42 | 54.62 | 72.67 | 77.11 | 75.60 | 77.04 |

Table 5.2: Steady-state model - the comparison of computational speed-up expressed as ratios of GPU and CPU calculation times.

### CUDA experiments

First, the parallel version of the model described by steady state equation (4.9) was evaluated only by CUDA framework using three different memory access scenarios. All tests were performed on a CPU and GPU separately. The experimental setup consisted of CPU Intel Core 2 Duo at 2.66 GHz. The selected graphic devices were GeForce 9600M GT as a representative of mobile devices, GeForce 8800 Ultra and new GeForce GTX280.

The model 9600M GT has 32 cores (laid out as 4 multiprocessors with 8 units) at a clock rate of 1.25 GHz. There are 8,192 registers available per block. Model 8800 Ultra has 128 cores (16 multiprocessors with 8 processors) with clock rate of 1.5 GHz. Again, there are 8,192 registers to be used for each block. GTX280, the last model used for evaluation, has 240 cores (30 multiprocessors with 8 processors). This time, there are 16,384 registers available for each thread block.

Three different approaches how to use the graphics memory were implemented. The first method (A) used only global memory of graphics card which is directly accessible by CPU. The second method (B) used shared memory of GPU to store the attributes of ODE system to have faster access from kernels running on GPU. The last method (C) uses texture memory with cached access for the equation values from a previous step (it is accessed 4 times during integration phase in each thread). Moreover, shared memory contained a number of auxiliary variables $k_i$ of Runge-Kutta method (3.17). The final results of the fastest method (C) are shown in table 5.2.

The experiments showed that in case of GTX280, the difference of calculation times between A, B and C methods - global vs. shared memory utilization, is not so big (48×

| Platform | Block size | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 4 | 8 | 16 | 32 | 64 | 128 | 256 |
| 9600M GT | 0.44 | 1.01 | 2.29 | 4.42 | 8.34 | 11.97 | 14.01 | 14.29 | 12.22 |
| 8800 Ultra | 2.30 | 5.24 | 11.59 | 21.64 | 39.92 | 56.49 | 68.86 | 65.01 | 57.11 |
| GTX280 | 4.67 | 10.59 | 22.61 | 40.57 | 74.08 | 105.67 | 115.58 | 115.87 | 115.61 |

Table 5.3: Time-dependent model - the comparison of computational speed-up expressed as ratios of GPU and CPU calculation times.

vs. 77×). In case of older graphics card the difference is quite huge - one order difference. The reason is the faster work with memory in case of GTX280 because 40% of the time was consumed by memory transfer. Moreover, GTX280 has the possibility to transfer data and calculate at the same time.

Similar set of experiments was done for time-dependent model (4.20). The results for 1048576 equations, which confirm the steady-state case, are shown in table 5.3.

### CUDA/OpenCL comparison experiments

The second set of experiments was done on the same version of equation. The testing application was written entirely in C++ language where the individual versions have been prepared for OpenCL and CUDA frameworks alike. As in the previous case, all experiments were performed on a CPU and GPU separately. The reference performance indicators are specified in case of single-thread application running on CPU Core 2 Duo at 2.267 GHz. The entire set of the following measurement is compared against these initial values.

All data processed by the computation kernel were read from a global memory on GPU card for each step of computation. However, efficient usage of local and private memories (as referred to in OpenCL specification) during computation process, together with overlay of asynchronous data transfers, helps to mitigate inherent latency.

Here, the experimental setup consisted of two CPUs: Intel Core 2 Duo at 2.267 GHz with 3 MB of L2 cache and Intel Core 2 Quad at 2.66 GHz with 6 MB of L2 cache. In addition, the following GPUs were used during experiments: GeForce GT9600M as a representative of mobile GPU, GeForce GTX285 as the high-end platform from nVidia and finally ATI HD5870.

The model GTX285 has 240 cores (30 multiprocessors with 8 processors) at a clock rate of approximately 1.5 GHz. This time, 16,384 registers are available for each thread block. Last example of GPU was ATI HD5870 which has 1,600 cores organized into 20 so called SIMD engines, where each of them works at a clock rate of 850 MHz.

The final experiment results are summarized in table 5.4 where the relative speed-ups against single threaded version running on Intel Core 2 Duo at 2.267 GHz are presented. There, an interesting fact could be noted with Dual Core CPU. When multiple threads were used in case of OpenCL the performance was worse than in single thread version. On the other hand, processing time on the quad core CPU is significantly lower. Furthermore, GPU platforms performed as expected. The column in table 5.4, which contains the values of relative speed-up in case of 512 threads per block, contains x mark for GeForce GT9600M. The reason is that application fails to be launched due to critical lack of resources.

| Platform | Block size | | | | |
|---|---|---|---|---|---|
| | 32 | 64 | 128 | 256 | 512 |
| OpenCL CPU (P8400, 2.26 GHz) | 0.91 | 0.89 | 0.90 | 0.87 | 0.45 |
| OpenCL CPU (Q9400, 2.66 GHz) | 2.09 | 2.09 | 2.06 | 2.12 | 0.54 |
| CUDA GPU (GT9600M) | 14.14 | 15.25 | 14.93 | 14.88 | 13.98 |
| OpenCL GPU (GT9600M) | 11.05 | 13.77 | 12.55 | 12.05 | x |
| CUDA GPU (GTX285) | 127.15 | 144.85 | 146.27 | 144.54 | 140.68 |
| OpenCL GPU (GTX285) | 122.87 | 136.67 | 136.93 | 130.97 | 126.31 |
| OpenCL GPU (HD5870) | 68.01 | 92.41 | 97.57 | 96.80 | 160.52 |

Table 5.4: The comparison of computational speed-up expressed as ratios of GPU and CPU calculation times. 262144 equations were calculated simultaneously in each step.

| Exp. num. | $h_s$ (m) | $H$ (m) | $L$ (m) | $a_\star$ (m s)$^{-1}$ | $w_\star$ (m s)$^{-1}$ | $K$ | $z_0$ (m) |
|---|---|---|---|---|---|---|---|
| 1 | 115 | 1980 | -46 | 0.37 | 1.70 | 0.4 | 0.6 |
| 2 | 115 | 1920 | -384 | 0.74 | 1.80 | 0.4 | 0.6 |
| 3 | 115 | 1120 | -108 | 0.39 | 1.10 | 0.4 | 0.6 |
| 4 | 115 | 390 | -173 | 0.39 | 0.74 | 0.4 | 0.6 |
| 5 | 115 | 820 | -577 | 0.46 | 2.50 | 0.4 | 0.6 |
| 6 | 115 | 1300 | -569 | 1.07 | 2.00 | 0.4 | 0.6 |
| 7 | 115 | 1850 | -136 | 0.65 | 2.10 | 0.4 | 0.6 |
| 8 | 115 | 810 | -72 | 0.70 | 2.10 | 0.4 | 0.6 |
| 9 | 115 | 2090 | -382 | 0.77 | 2.00 | 0.4 | 0.6 |

Table 5.5: The parameters of the performed experiments in Copenhagen [22].

### 5.1.3 Real models

In the next set of experiments, the real models of wind and dispersion were intended to use. The models were evaluated in correspondence with the Copenhagen experiments [21], [22] where sulphurhexafluoride substance was used. The tracer was released without buoyancy from a tower at a height of 115 meters and then collected 2-3 meters above ground-level at positions in up to three crosswind arcs of tracer sampling units, positioned 2-6 km from the point of release. Three consecutive 20 min averaged tracer concentrations were measured, allowing for a total sampling time of 1 hour. The site was mainly residential having a roughness length of 0.6 m.

There were 9 experiments performed in Copenhagen, in which all of the required parameters were measured. The all parameters of the experiments that were used for calculations are shown in table 5.5.

The experiments were done using the discretization scheme described by equation (4.41) with the appropriate boundary conditions (4.44). The spatial axis $z$ was discretized using the same step of size 8 m from ground to height of planetary boundary layer $H$ (m) defined individually for each experiment. The ODE was solved using the fourth-order Runge-Kutta method (3.17) with varying step size which were adopted in each step to meet the criteria of local error less than $1e^{-4}$. The concentration amount was collected from 2-3 meters during the real experiments and because the closest grid points in the experiments were in 0 m

and 8 m heights the concentration values in 2 m were interpolated.

It should be noted that MoL scheme represented by equation (4.41) contains the derivative of the diffusion function according to $z$. It can be obtained either in exact form or, if impossible, as numerical approximation. The treatment in concrete model cases will be shown in the following subsection.

### Wind and turbulent parametrization

The wind speed in both profiles used in the experiments is dependent on height $z$ variable and other meteorological parameters measured at site. The two concrete profiles in case of the height of the unstable boundary layer $H = 1980$ m (Copenhagen experiment number 9) are shown in figure 5.2.

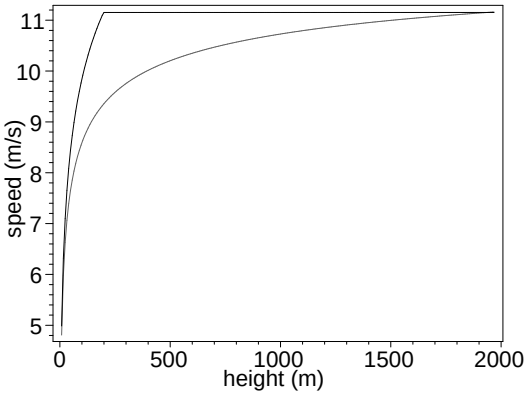



Figure 5.2: The real wind profiles used in experiments - equation (2.29) (black) and equation (2.34) (grey).

The turbulent parametrization was done using three models. First two models where the diffusion is dependent on height $z$ [60], [56] are shown on the left side in figure 5.3. The third model dependent on height $z$ and downwind distance from the source $x$ is shown on the right side in figure 5.3.

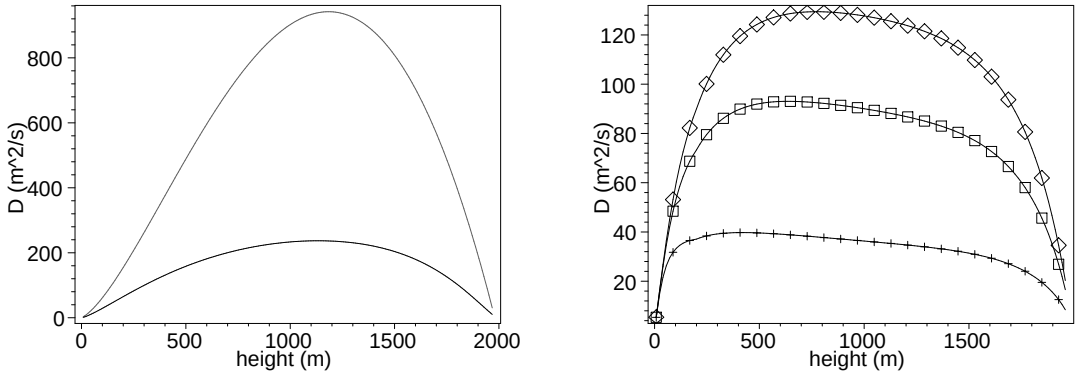



Figure 5.3: The real turbulent profiles used in experiments. The turbulent profiles of equations (2.35) and (2.34) are shown on the left (black and grey). The turbulent profile of equation (2.36) for the downwind distances of 1000 m (cross), 3000 m (box) and 5000 m (diamond) are shown on the right side.

| Scheme | $L_1$-norm error | $L_2$-norm error | $L_\infty$-norm error |
|---|---|---|---|
| MoL | $0.840 \cdot 10^{-1}$ | $0.136 \cdot 10^{-1}$ | 0.274 |

Table 5.6: $L_n$-norm error measures of performed experiments with Wortmann turbulent parametrization where analytical solution is known.

As it was outlined above, the derivatives of diffusion equations have to be known in order to solve the numerical schemes. After some calculations the derivative of function 2.35 has the form

$$\begin{aligned}\frac{\partial D}{\partial z} =& 0.07\overline{3}\frac{\left(1-\frac{z}{H}\right)^{1/3}\left(1-\exp(-4z/H)-0.0003\exp(8z/H)\right)w_\star}{\left(\frac{z}{H}\right)^{2/3}} - \\ & 0.07\overline{3}\frac{\left(\frac{z}{H}\right)^{1/3}\left(1-\exp(-4z/H)-0.0003\exp(8z/H)\right)w_\star}{\left(1-\frac{z}{H}\right)^{2/3}} + \\ & 0.22\left(\frac{z}{H}\right)^{1/3}\left(1-\frac{z}{H}\right)^{1/3}\left(4\frac{\exp(4-z/H)}{H}-0.0024\frac{\exp(8-z/H)}{H}\right)w_\star H.\end{aligned} \tag{5.1}$$

Similarly the algebraic form of derivative of equation (2.34) can be found

$$\begin{aligned}\frac{\partial D}{\partial z} &= Ka_\star\left[\frac{1-z/H}{1+9.2z/L}-\frac{z}{H(1+9.2z/L)}-9.2z\frac{1-z/H}{(1+9.2z/L)^2L}\right] \text{ for } H/L>0 \\ \frac{\partial D}{\partial z} &= Ka_\star\left[\sqrt{1-13z/L}\left(1-\frac{2z}{H}\right)-\frac{13z}{2}\frac{1-z/H}{\sqrt{1-13z/L}L}\right] \text{ for } H/L\le 0.\end{aligned} \tag{5.2}$$

The derivative of last turbulent diffusion equation (2.36) is hardly possible to get in algebraic form. Therefore, the derivative was approximated by the central difference defined by equation (3.9) which showed to have enough accuracy in the experiments.

**Results**

The first experiment was done using the the wind and turbulent parametrization by Wortmann et al. [60] where the analytical solution is known. The analytical method which basic description is in section 4.1.4 was implemented. The results from numerical model were evaluated and the corresponding $L_n$-norm error measures are shown in table 5.6. There, the very good correspondence between the two solutions could be seen.

The rest of experiments were done only using numerical solution of the defined models. The left side in figure 5.4 shows the comparison between predicted and measured crosswind-integrated concentrations $C$ (kg m$^{-2}$) in all experiment cases. For clarity, the concentrations are normalized according to source term and divided by $10^4$: $C_{o|p} = 10^{-4}C/Q$, with units of (kg m$^{-2}$)/(kg s$^{-1}$) = s m$^{-2}$. The ideal state would be if the points lie on the middle line. The other two lines border the space of factors 0.5 and 2 (see equation (3.68)) and it is seen that all the predicted values lie in this range.

The second experiment with the same data but different wind and diffusion models from [56] was performed. The overall results are shown in figure 5.4 on the right side. The

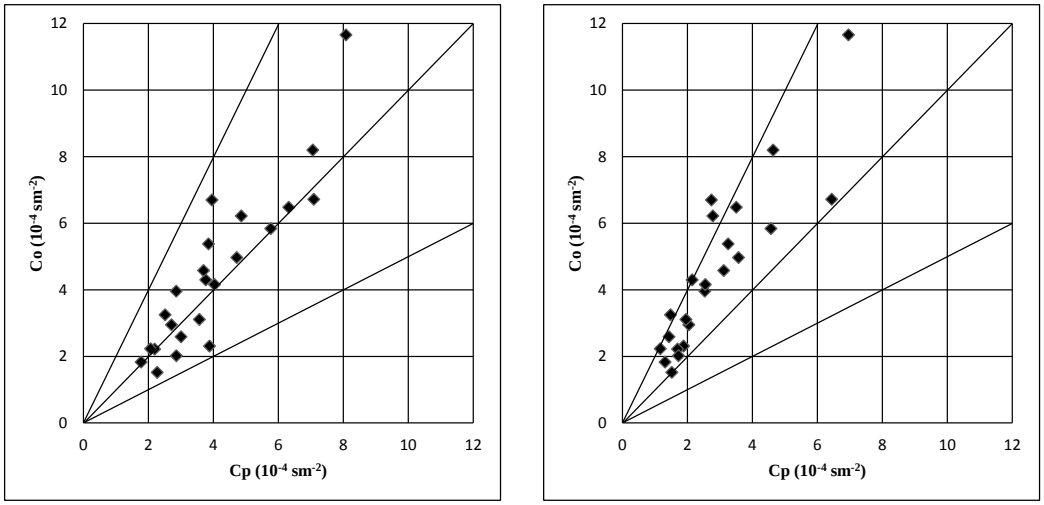



Figure 5.4: The comparison of the measured concentrations ($C_o$) and the predicted concentrations ($C_p$) using the dispersion parametrization by Wortmann [60] (left) and Ulke [56] (right).

| Model | NMSE | COR | FA2 | FB | FS |
|---|---|---|---|---|---|
| Analytical (W) | 0.08 | 0.90 | 1.00 | 0.11 | 0.32 |
| MoL (W) | 0.08 | 0.90 | 1.00 | 0.10 | 0.33 |
| MoL (U) | 0.34 | 0.89 | 0.83 | 0.45 | 0.44 |
| MoL (U/W) | 0.21 | 0.92 | 1.00 | 0.34 | 0.42 |
| MoL (D) | 0.08 | 0.88 | 1.00 | 0.08 | 0.23 |

Table 5.7: The statistical indexes values of all performed experiments with real wind/turbulent parametrization.

results are a bit worse than in the previous case, however, this cannot be the final decision because the number of experiments is relatively small. Moreover, there are cases when Ulke's parametrization is better than Wortmann's.

For the third set of experiments the turbulent parametrization dependent on height and downwind distance was chosen. This overall model is more accurate than the ones presented before. The results are shown in figure 5.5 on the right side. The parametrization of the last experiment is a combined one - the wind is from [60] and turbulent from [56] and which results are shown on the right side in figure 5.5.

The measured statistical indices are shown in table 5.7. In the first two rows the statistics for the analytical and numerical solutions of Wortmann's model are stated showing almost the same accuracy. The solutions to models containing the wind and/or turbulation parametrization proposed by Ulke (labelled with U and U/W) indicate worse results. The best results, shown in the last row, were achieved by using turbulent parametrization by Degrazia [16]. As a conclusion, it can be noted that the model results correspond to the observed concentration levels very well.

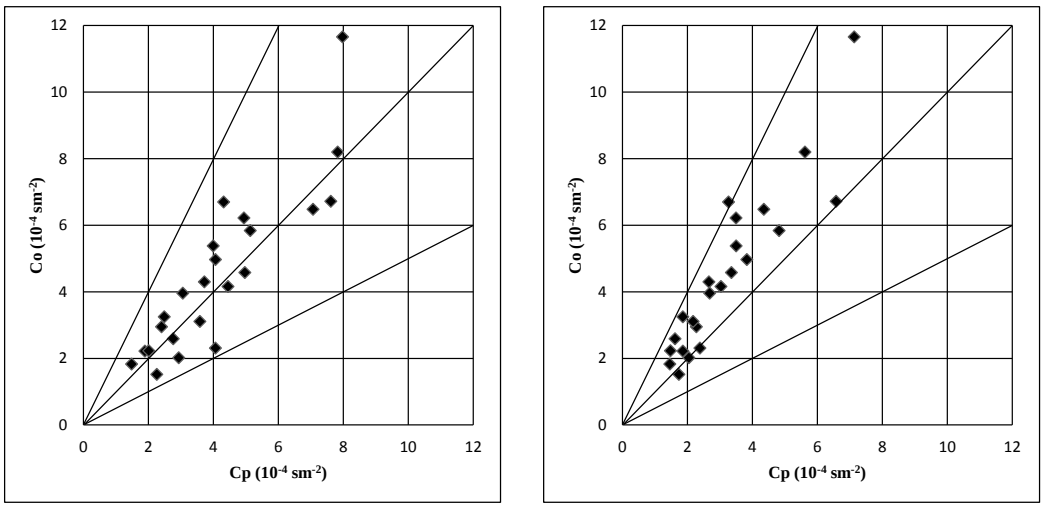



Figure 5.5: The comparison of the measured concentrations ($C_o$) and the predicted concentrations ($C_p$) using the dispersion parametrization by Degrazia [16] (left) and the combined model, wind by Wortmann, turbulent by Ulke (right).

## 5.2 ELLAM framework

This section describes the experiments done with the original ELLAM method inspired by [33] and its modified version using adaptation techniques presented in section 4.2.2. The first set of experiments was done for pure advection cases using artificial rotating wind model. Here the exact solution is known and the computed values are directly evaluated. The second set of experiments was done for artificial divergent wind model where the global mass conservation was studied. The last set of experiments is dedicated to advection-diffusion phenomena with source term where the real models of advection and diffusion terms are used in order to show the suitability of the framework in this application.

### 5.2.1 Rotation wind

Rotation wind tests are quite common techniques to show the performance of the numerical schemes solving pure advection equations. Here, the domain is in two dimensions and has square shape. The wind rotates at a constant rate during the whole simulation. Under these conditions, it is obvious that concentration profile of any initial shape has to remain the same at the end of each simulation if the concentration is zero at the domain boundaries and thus it is not out-flowed away.

#### Configuration

As it is stated further, different initial concentration shapes cause smaller or bigger problems to the tested schemes which were used. The relatively easy shape is of non-steep cone. The moderate difficulties are caused by the cylindrical shape where vertical gradients are presented but horizontally the shape is smooth. The hardest shape also used in these tests is a slotted cylinder, i.e. cylinder from which the box shape is subtracted. Thus the shape has both vertical and horizontal discontinuities. The initial shape profiles are shown if figure

| Scheme | MassE | RMSE | PeakE | DistrE |
|---|---|---|---|---|
| Walcek | $0.118 \times 10^{-10}$ | $0.120 \times 10^{-1}$ | $0.414 \times 10^{-1}$ | $0.311 \times 10^{-1}$ |
| ELLAM | $0.905 \times 10^{-4}$ | $0.221 \times 10^{-2}$ | $0.607 \times 10^{-1}$ | $0.313 \times 10^{-2}$ |
| Adapt. ELLAM | $0.142 \times 10^{-4}$ | $0.321 \times 10^{-2}$ | $0.755 \times 10^{-1}$ | $0.206 \times 10^{-3}$ |

Table 5.8: The mean errors of all performed experiments with cone initial profile.

5.6 where the shapes are projected in the height properly scaled with respect to horizontal axes.

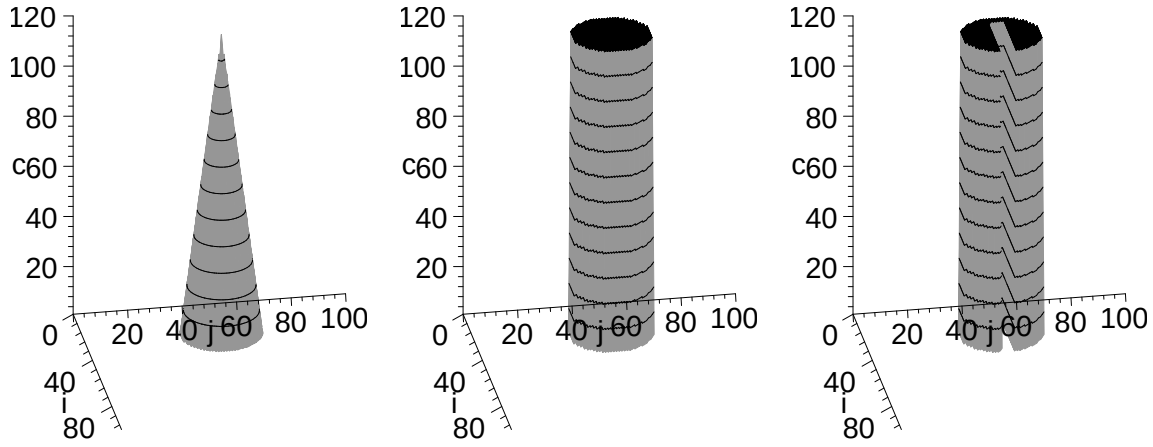



Figure 5.6: The initial shape profiles used in artificial tests for pure advection equations - cone (left), cylinder (center) and slotted cylinder (right).

The experiment settings were as follows. All tests were done in a squared space which was divided into $100 \times 100$ points. The diameters of the initial shapes were set to 30 points for all tests. The time steps were set to 360 per one rotation, i.e., the Courant number was less than one in case of Walcek algorithm. On the other hand the time step was set to $8^{th}$ and $24^{th}$ multiple of Walcek setting in case of original ELLAM. The step size of ELLAM algorithm was chosen in this way to reach the approximately same calculation time as in case of Walcek scheme. Obviously, the adapt version of ELLAM sets its step size to necessary values to reach the (sub-)optimal time stepping to gain significantly smaller oscillations and as small numerical diffusion as possible.

All the results in rotating experiments were evaluated using the error measurements defined in section 3.2.6 in its *Concentration profile preserving* subsection.

### Cone profile

The tests with a cone shape profile were done for short-, mid- and long-term simulations represented by 1, 6 and 60 rotations. The results at the end of the appropriate simulations are shown in table 5.8. Obviously, all schemes preserve mass very well. Since original ELLAM performs with cone profile very well and adapt version of ELLAM adds some diffusivity the peak error is smaller in case of original ELLAM scheme. RMSE and DistrE are smaller in both ELLAM schemes.

The differences among the three studied numerical schemes (Walcek, ELLAM, Adapt. ELLAM) can be seen in detail in figure 5.7 in case of 60 rotations. Walcek scheme suffers

| Scheme | MassE | RMSE | PeakE | DistrE |
|---|---|---|---|---|
| Walcek | $0.151 \times 10^{-9}$ | $0.712 \times 10^{-1}$ | 0.0 | 0.162 |
| ELLAM | $0.511 \times 10^{-4}$ | $0.483 \times 10^{-1}$ | $0.889 \times 10^{-1}$ | $0.523 \times 10^{-1}$ |
| Adapt. ELLAM | $0.507 \times 10^{-4}$ | $0.537 \times 10^{-1}$ | $0.386 \times 10^{-1}$ | 0.117 |
| Walcek | $0.971 \times 10^{-11}$ | $0.822 \times 10^{-1}$ | 0.0 | 0.232 |
| ELLAM | $0.503 \times 10^{-4}$ | $0.591 \times 10^{-1}$ | 0.153 | $0.998 \times 10^{-1}$ |
| Adapt. ELLAM | $0.327 \times 10^{-3}$ | $0.497 \times 10^{-1}$ | $0.454 \times 10^{-1}$ | $0.991 \times 10^{-1}$ |

Table 5.9: The mean errors of all performed experiments with cylinder (first three rows) and slotted cylinder (last three rows) initial profiles.

a bit from mass shifting during long-term experiments. This is most probably caused by applied dimensional splitting approach here - the calculations are done sequentially one dimension after the other. The similar observations have been presented also by the authors [58].

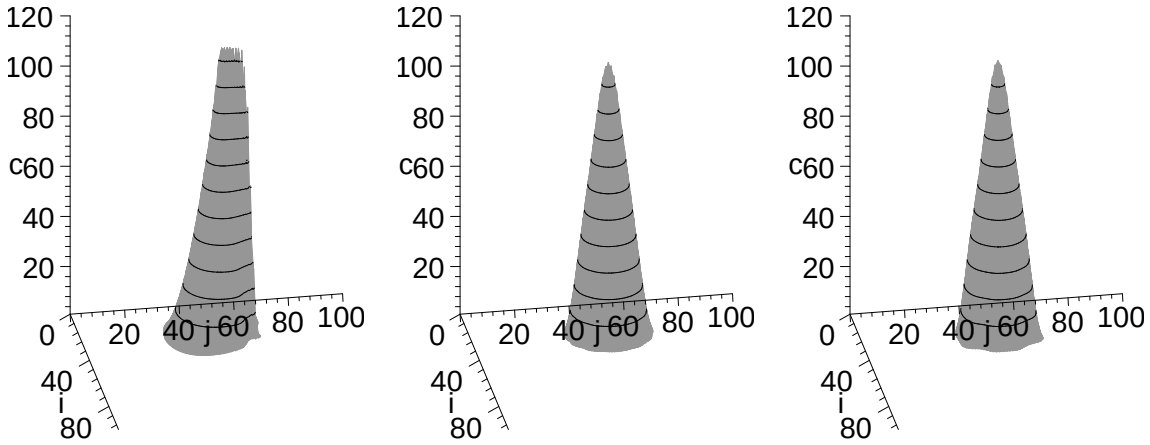



Figure 5.7: The results of simulation with 60 rotations and cone initial profiles for Walcek (left), ELLAM (center) and Adapt. ELLAM (right) schemes.

### Cylinder-based profiles

The experiments with the cylinder-based initial profiles were done under the same conditions as in the previous case. The number of rotations was set to 1, 6 and 60. The results for cylinder and slotted cylinder are shown in table 5.9. The results show again the similar very good mass preserving of all numerical schemes, Walcek has the smallest peak error and ELLAM and Adapt. ELLAM have significantly smaller RMSE and DistrE. Moreover, Adapt. ELLAM has significantly lower PeakE than original ELLAM scheme.

The results of profile shapes after 60 rotations of all three schemes are shown in figure 5.8. Surprisingly, Adapt. ELLAM calculated more diffused scheme than its original form. This is probably caused by quite large number of deduced time step during the experiment (694 against 35 in case of slotted cylinder) together with artificial diffusion which was added to maintain less oscillations. These are not presented at all in this case and the shape with its error measurements are comparable to Walcek scheme.

The same figure shows also the results of experiments with the slotted cylinder.In this case, there could be seen that Adapt. ELLAM reached the best results. This comes from the fact that very small number of time steps were deduced to perform accurate, almost oscillation-free simulation.

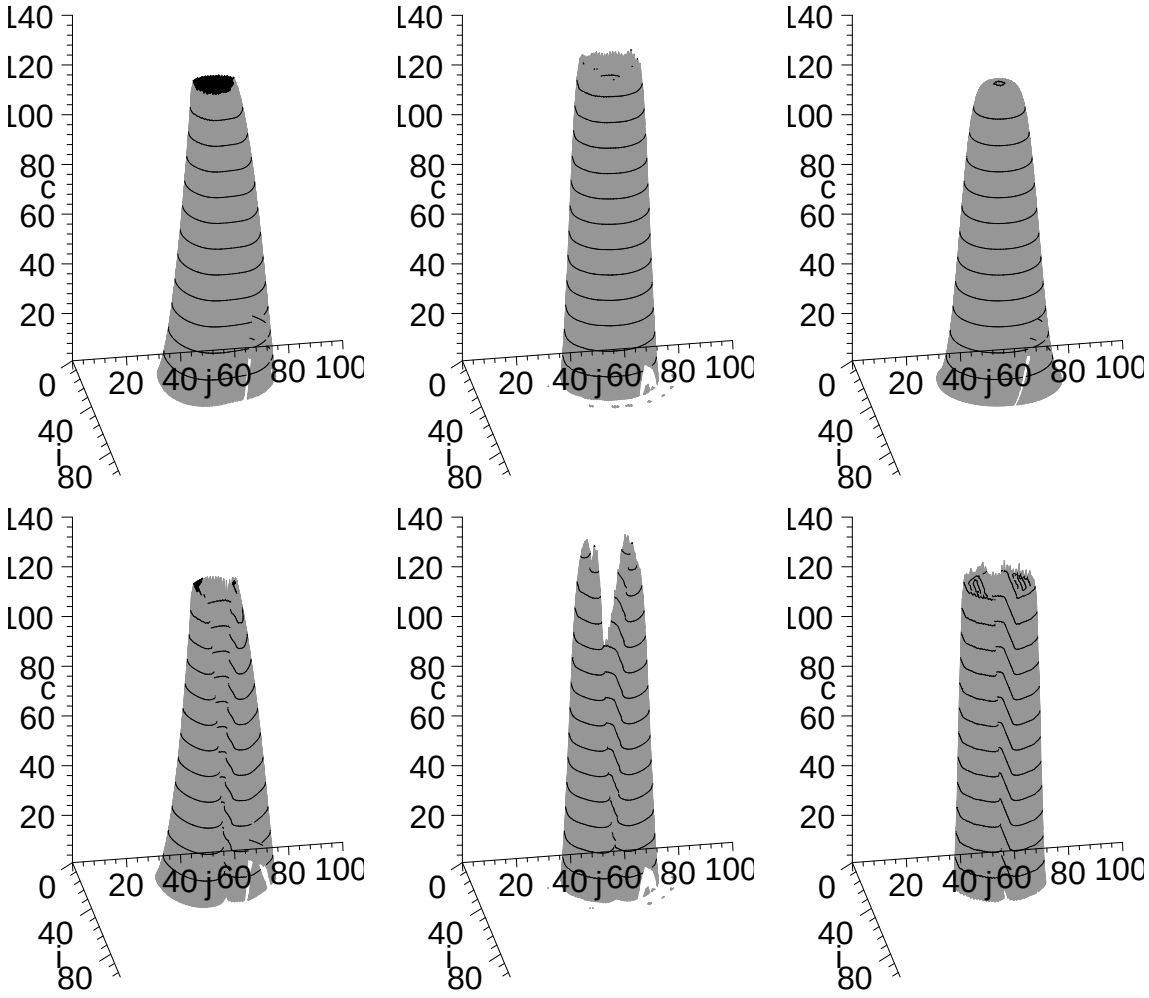



Figure 5.8: The results of simulation with 60 rotations, and cylinder (the images in the first row) and slotted cylinder (the images in the second row) initial profiles for Walcek (left), ELLAM (center) and Adapt. ELLAM (right) schemes.

### 5.2.2 Divergent wind

The last set of artificial experiments was done with divergent wind model that was presented also in [58]. The wind speeds along $x$ and $y$ axes are computed using the following equations

$$a_x = \sin\left(\frac{\pi i}{25}\right)\sin\left(\frac{\pi j}{25}\right), \tag{5.3}$$

$$a_y = \cos\left(\frac{\pi(i+d_i)}{25}\right)\cos\left(\frac{\pi(j+d_j)}{25}\right), \tag{5.4}$$

where $i$ and $j$ are cell indexes along $x$ and $y$ axes and $d_i$ and $d_j$ lying in $[0, 1]$ interval are the displacements of the wind in $y$ direction. The part of the velocity field is shown in figure 5.9 where also the base of the used initial profiles is displayed. It is presented there that the wind blows in circles within squares of size $25 \times 25$ cells. The maximum wind speed was set to 10 and it is reached at edges of the squares. The zero velocity of the wind is presented in the middle of the squares.

At long times after initialization, the concentration distribution becomes sheared within the swirls into infinitesimal „curtains“ or sheets which wrap around one another while becoming thinner, and therefore are not resolved by a numerical grid mesh of $(100\Delta x)^2$ [58]. Therefore, if the exact solution is averaged over $(\Delta x)^2$ areas at long times after initialization, the tracer should become uniformly mixed along streamlines of the swirls, and have a ring-like structure within each vortex that contained any of the tracer initially.

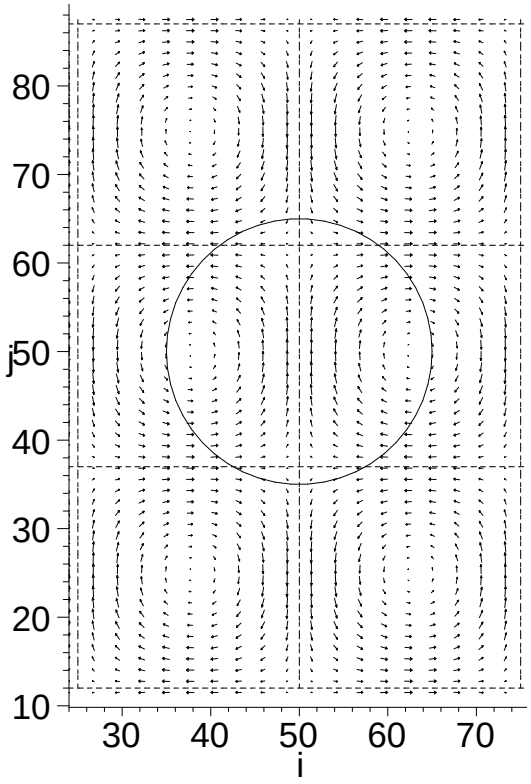



Figure 5.9: The velocity field of divergent wind used in experiments displayed with the base of the used concentration profile shapes.

The first set of experiments were done according to one performed in [58] with cone initial profile which center was placed at $[50, 50]$, i.e. in the middle of the domain $100 \times 100$ as it is shown in figure 5.9. The element size was set to $40000 \times 40000$ m$^2$ and the time step was set to 2637.6 s in case of Walcek method. Two cases of ELLAM were tested as before - original ELLAM version according to [33] with 24 time bigger time step than Walcek and Adapt. ELLAM which used adaptive time steps in combination with adaptive diffusion.

The results after 8000 Walcek time steps are shown in figure 5.10 where the range of concentration is from 0.1 to 120 (concentrations near zero are not shown for clarity). It is obvious that only Walcek method reached the criteria that pollutant should not be advected to other wind squares than initial profile was presented in. The pollutant is spread

most in case of Adapt. ELLAM which is most probably caused by the artificial diffusion added to reduce oscillations. However, the mass is better preserved here than in case of original ELLAM as it will be shown further. On the other hand, Walcek method shows an asymmetry in final pollutant distribution. This is most probably caused by dimension splitting approach.

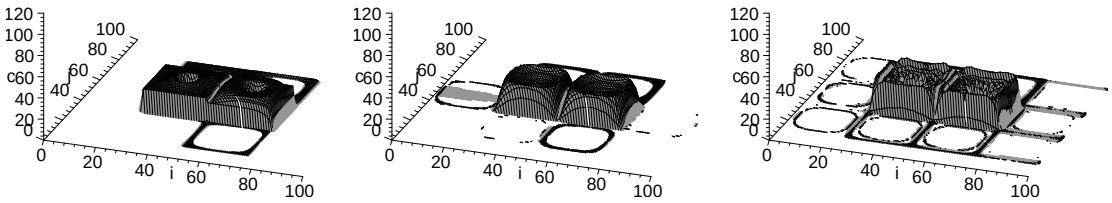



Figure 5.10: The results of simulation with divergent wind after 8,000 iterations of Walcek method; Walcek (left), ELLAM (center) and Adapt. ELLAM (right) schemes.

Previously, the wind velocity field was arbitrarily shifted ($d_i = -0.5$ and $d_j = 0.5$ in equation (5.4)) to have the edges of the wind squares exactly at cell boundaries. The detail of this case is shown in figure 5.11 on the left side where the wind square corner has $[12, 50]$ coordinates. The next set of experiments were done with the same divergent wind, however, with the $j$ square edges placed to the cell center ($d_i = 0$ and $d_j = 0$ in equation (5.4)). The case is shown in figure 5.11 on the right side.

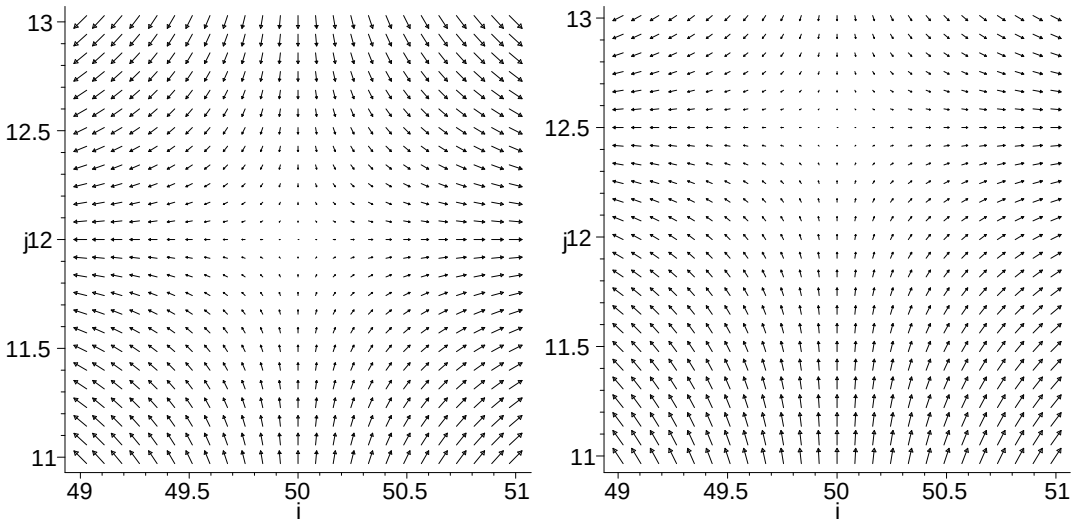



Figure 5.11: The velocity field of divergent wind used in experiments.

The results of the experiments after 8000 time steps of Walcek method are shown in figure 5.12. Here, in Walcek case the big amount of pollutant out-flowed from the domain. Original ELLAM performed better and Adapt. ELLAM kept the most of the pollutant correctly in the two wind squares. The MassE measures that confirm this will be presented further.

### Summary

As a summary of this section, the results of relative MassE measured in each experiment are presented. The same experiments using divergent wind were done also with cylinder

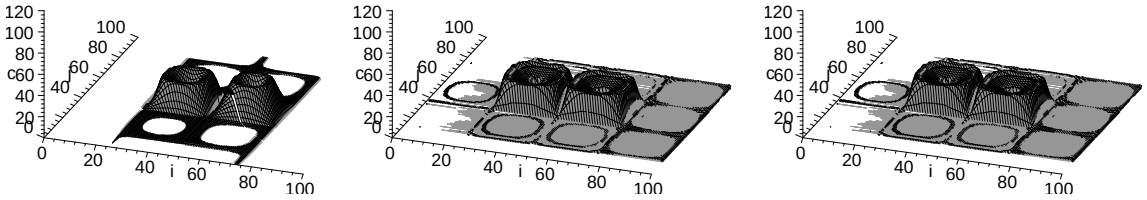


Figure 5.12: The results of simulation with shifted divergent wind after 8000 iterations of Walcek method; Walcek (left), ELLAM (center) and Adapt. ELLAM (right) schemes.

| Wind model | Initial profile | Scheme | MassE |
|---|---|---|---|
| Original wind | Cone | Walcek | $0.339 \times 10^{-3}$ |
| Original wind | Cone | ELLAM | 0.132 |
| Original wind | Cone | Adap. ELLAM | $0.384 \times 10^{-1}$ |
| Original wind | Cylinder | Walcek | $0.641 \times 10^{-2}$ |
| Original wind | Cylinder | ELLAM | 0.157 |
| Original wind | Cylinder | Adap. ELLAM | 0.118 |
| Original wind | Slotted cylinder | Walcek | $0.658 \times 10^{-2}$ |
| Original wind | Slotted cylinder | ELLAM | 0.173 |
| Original wind | Slotted cylinder | Adap. ELLAM | 0.123 |
| Shifted wind | Cone | Walcek | 0.386 |
| Shifted wind | Cone | ELLAM | 0.237 |
| Shifted wind | Cone | Adap. ELLAM | 0.122 |
| Shifted wind | Cylinder | Walcek | 0.381 |
| Shifted wind | Cylinder | ELLAM | 0.256 |
| Shifted wind | Cylinder | Adap. ELLAM | 0.209 |
| Shifted wind | Slotted cylinder | Walcek | 0.394 |
| Shifted wind | Slotted cylinder | ELLAM | 0.277 |
| Shifted wind | Slotted cylinder | Adap. ELLAM | 0.215 |

Table 5.10: MassE measures of all performed experiments with cone, cylinder and slotted cylinder initial profiles. First 9 rows show the results for original wind model, the last 9 rows show the results where shifted wind model was used.

and slotted cylinder initial profiles with the same base diameter. In these cases the two divergent winds were also used - the original one and the shifted one.

The final results in the form of relative MassE are shown in table 5.10. It is clearly visible that Walcek has the best results in case of original divergent wind. On the other hand, it shows its flaws when shifted wind profile is used where it has bigger relative MassE. It is also evident that Adapt. ELLAM has smaller error than original ELLAM in all performed experiments.

### 5.2.3 Real advection-diffusion models

The last set of experiments of ELLAM framework was done for real advection-diffusion models that were also used in the implementation of MoL method. As the input, the data from nine experiments performed in Copenhagen was used (see table 5.5). The wind parametrization from [60] (see equation (2.29)) was used and the dispersion parametrization

| Model | NMSE | COR | FA2 | FB | FS |
|---|---|---|---|---|---|
| MoL (W) | 0.08 | 0.90 | 1.00 | 0.11 | 0.32 |
| MoL (U) | 0.21 | 0.91 | 1.00 | 0.33 | 0.40 |
| ELLAM (W) | 0.14 | 0.83 | 0.96 | 0.18 | 0.32 |
| ELLAM (U) | 0.30 | 0.82 | 0.87 | 0.34 | 0.48 |
| Adapt. ELLAM (W) | 0.15 | 0.82 | 0.96 | 0.19 | 0.33 |
| Adapt. ELLAM (U) | 0.29 | 0.82 | 0.83 | 0.37 | 0.48 |

Table 5.11: The statistical indexes values of all performed experiments with real wind/turbulent parametrization.

was used by [60] (see equation (2.35)) and [56] (see equation (2.34)).

The testing models were similar as the ones used in section 5.1.3. The models used here were MoL, ELLAM and Adap. ELLAM. The form of the MoL designed in section 4.1.4 with up-winding was used during testing. ELLAM and Adapt. ELLAM models had the same form as the ones with the same name used previously in this section extended with point source term. All of the models were time-dependent with steady-state point source placed at the coordinates as it was in case of performed Copenhagen experiments. Therefore, the condition to stop the simulation was defined in the following way. The simulation stops when steady concentration level is reached in all places inside the domain.

The domain was discretized to 10 m $\times$ 10 m squares in vertical and horizontal directions. It means that for instance the space of Copenhagen experiment No. 1 with collecting distance of 1900 m from the source and mixing atmospheric height of 1980 m was discretized into 199 points in case of MoL or 198 cells in case of ELLAM methods. The time stepping was set to 4 s at the beginning which remains the same only in case of ELLAM method, the other two methods used adaptive time stepping as described before.

The important fact is that the accurate point source could be only used in MoL approach since in ELLAM case the average concentrations in the cells are used and thus the source was actually area-based with 10 $m^2$ and thus this condition differs in the models. This influenced the results between MoL and ELLAM. The results are also influenced by the fact that there were no boundary condition terms (Neumann near the ground) implemented in case of ELLAM methods.

The results of performed experiments are shown in table 5.11. The results confirmed the assumptions that the MoL and ELLAM computed concentrations differ. However, the two ELLAM methods agreed with each other very well.

There was also an interesting observation in terms of the calculation speed. The overall time to calculate the final results of all experiments differed in all methods. The fastest was ELLAM (3.46 hours), Adapt. ELLAM was the second fastest (3.89 hours) and the slowest was MoL (5.33 hours). Adapt. ELLAM was slightly slower because of the extra calculations needed for the time-stepping and diffusion adaptations. MoL used adaptive time-stepping in order to maintain accuracy and finally leaded to the big calculation time.

From the results, it can be deduced that there is no advantage of Adapt. version of ELLAM. The fact that samples of concentrations were collected from far distances from the source leads to the very similar results of the two ELLAM methods. However, the difference of the methods is seen near the source of the pollution where due to relative big time-steps the oscillation of ELLAM appears. An example is shown in figure 5.13 on left where it is clearly visible that original ELLAM could suffer from oscillations and that better results

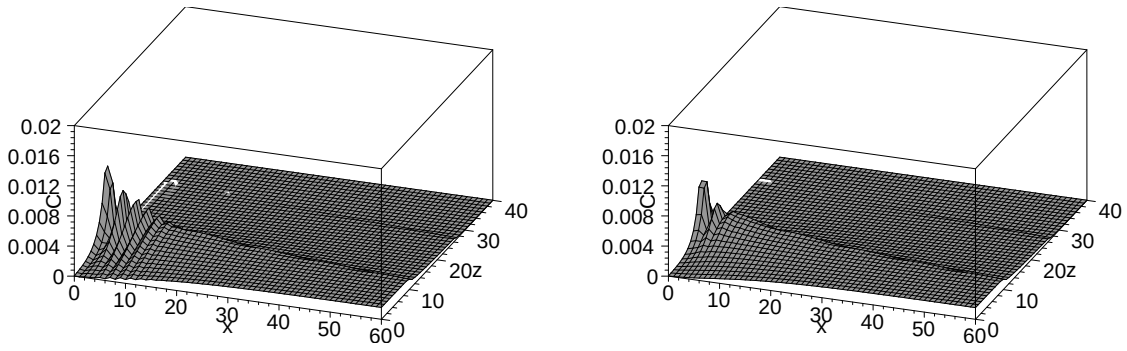



Figure 5.13: The results of simulation of Copenhagen experiment no. 9. The details of concentration near the source are shown for ELLAM (left) and Adap. ELLAM (right) methods. The units are $10^1$ m in case of $x$ and $z$ axes, and $10^{-4}$ sm$^{-2}$ in case of $C$ axis.

are provided by Adapt. ELLAM although it does not compensate oscillations completely. The problem here is that not all the types of oscillations are detected due to presence of the point source and thus the concentration nearby is the highest.

# Chapter 6

# Conclusion

This dissertation thesis deals with numerical solutions of advection-diffusion equation describing the pollutant dispersion in planetary boundary layer. The work contains studies of the two concrete methods, method of lines and ELLAM. It was shown in artificial and real models that both methods are suitable for the models of local/urban scales.

## 6.1 Used methods and achieved results

The numerical solution based on method of lines was firstly derived for steady-state problem where the source term was constant during the simulation and thus the problem led to a stationary solution. Next, the basic stability analysis was done on time-dependent variant were the time and space steps limits were defined to maintain the stability of the method. The contribution of this thesis with respect to method of lines lies in *design and testing of its parallel version* using OpenCL and CUDA platforms (see sections 4.1.3 and 5.1.2 for details). It was shown that graphic cards are very good for general purpose computing using MoL approach with $4^{th}$-order Runge-Kutta method even in case of their former versions which used only 32bit floating point numbers.

The second part of MoL application was dedicated to real model parametrization using several existing wind and dispersion models. The numerical models in the steady-state and time-dependent forms were validated using known analytical solution of one particular case plus using the results from performed experiments done in Copenhagen. The purpose of such studies was to verify that quite simple MoL is suitable to solve the pollution models for local/urban scale which was confirmed for Copenhagen using single CPU core.

Quite young ELLAM method was the second topic to study in this thesis. The goal was to find out if it is suitable to effectively solve the advection part or whole advection-diffusion equation and to compare it with the used state of the art methods. The certain form of ELLAM was implemented for structured grid which was directly compared in various experiments of pure advection problems with Walcek's method. The result showed that ELLAM performed very well, especially its adaptive version which prevents the oscillations the original form suffered from. Although there exist several methods how to overcome oscillations, the proposed modifications have the advantage that they are in general applicable to different forms of ELLAM and they are simple to implement. The designed *Adaptive ELLAM* can be considered as the second contribution of this work (see sections 4.2.2 and 5.2 for details).

ELLAM method was also tested on data from Copenhagen experiments using the two

concrete real wind and the dispersion parametrizations. It was shown that the ELLAM and Adaptive ELLAM were able to solve the cases in shorter time than MoL approach with up-winding. Moreover, the results from Adaptive ELLAM had better quality because of the reduction of artificial oscillations.

## 6.2 Possible future work

There are many possibilities how to extend the methods and continue with the research in the future. The designed MoL approach and its parallel version can be extended with the higher-order methods to approximate the space derivatives more accurately and higher-order method to solve its system of ordinary differential equations. However, the attention needs to be paid here to keep the accuracy and stability. Next, the parallel version of MoL can be tested on more recent GPU and CPU HW that support CUDA/OpenCL frameworks or it can be quite easily extended for computations on supercomputers that are dedicated for the purpose of pollution prediction. Although MoL is quite old and outdated method it is is still very popular because of its simplicity and recently because of big expansion of GPU for general purpose computing and the multi-core architectures in general. The latter can successfully contribute to better accuracy by using significantly more equations and still keep the similar computational time.

A relative simple form of ELLAM was used in this thesis. In literature, there already exist several extensions. ELLAM was successfully extended to three dimensional problems and with unstructured grids. Also, the techniques to incorporate different types of boundary conditions and to incorporate reaction term were proposed. These extended versions combined with proposed adaptation techniques could be quite interesting for future studies. The interesting would be also to compare advection-diffusion variant of ELLAM with the approaches that are really used in practice like in AURORA where advection part is solved by Walcek's method and diffusion part with source term by implicit Crank-Nicolson scheme.

Similarly to MoL, also ELLAM method has a big potential for distributed computing. Namely the assembling of the matrices, characteristic tracking or solving of the final system of algebraic equations are very good candidates.

For advection part of the ADE equation, one could also consider WENO method which is also presented in the thesis. Although, the variants presented here did not achieve better results than Walcek's method it could not hold for its higher order variants.

# Bibliography


[1] IEEE standard for floating-point arithmetic. *IEEE Std 754-2008*, pages 1–70, Aug 2008.

[2] D. Barton, I. M. Willers, and R. V. M. Zahar. The automatic solution of systems of ordinary differential equations by the method of taylor series. *The Computer Journal*, 14(3):243–248, 1971. ISSN 1460-2067.

[3] P. Binning and M. A. Celia. A forward particle tracking Eulerian–Lagrangian Localized Adjoint Method for solution of the contaminant transport equation in three dimensions. *Advances in Water Resources*, 25(2):147 – 157, 2002. ISSN 0309-1708.

[4] J. Blazek. *Computational Fluid Dynamics - Principles and Applications*. Elsevier Science, 2001. ISBN 978-0-080-54554-7.

[5] A. Bott. Monotone flux limitation in the area-preserving flux-form advection algorithm. *Monthly Weather Review*, 120(1):2592 – 2602, 1992. ISSN 1520-0493.

[6] T. Brandvik and G. Pullan. Acceleration of a 3D Euler solver using commodity graphics hardware. In $46^{th}$ *AIAA aerospace sciences meeting and exhibit*, page 607. Curran Associates, Inc., 2008. ISBN 978-1-605-60201-1.

[7] P. J. H. Builtjes. Major twentieth century milestones in air pollution modelling and its application. In Sven-Erik Gryning and FrancisA. Schiermeier, editors, *Air Pollution Modeling and Its Application XIV*, pages 3–16. Springer US, 2001. ISBN 978-0-306-46534-5.

[8] J. C. Butcher. *Numerical Methods for Ordinary Differential Equations*. Wiley, $2^{nd}$ edition, 2008. ISBN 978-0-470-72335-7.

[9] M. Cassol, S. Wortmann, and U. Rizza. Analytic modeling of two-dimensional transient atmospheric pollutant dispersion by double GITT and Laplace transform techniques. *Environmental Modelling & Software*, 24:144–151, 2009. ISSN 1364-8152.

[10] M. A. Celia, T. F. Russell, I. Herrera, and Ewing R. E. An Eulerian-Lagrangian localized adjoint method for the advection-diffusion equation. *Advanced Water Resources*, 13:187–206, 1990. ISSN 0309-1708.

[11] T. J. Chung. *Computational Fluid Dynamics*. Cambridge University Press, 2002. ISBN 978-0-521-59416-5.

[12] NVIDIA Corporation. *NVIDIA CUDA C Programming Guide*, February 2014.

[13] A. Daly and P. Zannetti. Air pollution modeling–An overview. In P. Zannetti, D. Al-Ajmi, and S. Al-Rashied, editors, *Ambient air pollution*. The Arab School for Science and Technology (ASST) and The EnviroComp Institute, 2007.

[14] A. W. Date. *Introduction to Computational Fluid Dynamics*. Cambridge University Press, 2005. ISBN 978-0-521-85326-2.

[15] K. Datta, M. Murphy, V. Volkov, S. Williams, J. Carter, L. Oliker, D. Patterson, J. Shalf, and K. Yelick. Stencil computation optimization and auto-tuning on state-of-the-art multicore architectures. In *Proceedings of the 2008 ACM/IEEE conference on Supercomputing*, page 4. IEEE Press, 2008. ISBN 978-1-424-42835-9.

[16] G. A. Degrazia, D. M. Moreira, C. R. J. Campos, J. C. Carvalho, and M. T. Vilhena. Comparison between an integral and algebraic formulation for the eddy diffusivity using the Copenhagen experimental dataset. *Il Nuovo Cimento*, 25:207–218, 2002. ISSN 0390-5551.

[17] G. A. Degrazia, D. M. Moreira, and M. T. Vilhena. Derivation of an eddy diffusivity depending on source distance for vertically inhomogeneous turbulence in a convective boundary layer. *Journal of Applied Meteorology*, 40(7):1233–1240, 2001. ISSN 1520-0450.

[18] D. L. Ermak. An analytical model for air pollutant transport and deposition from a point source. *Atmospheric Environment (1967)*, 11:231–237, 1977. ISSN 0004-6981.

[19] F. A. Gifford. Turbulent diffusion-typing schemes: a review. *Nuclear Safety*, 17(1): 68–86, Jan 1976.

[20] Khronos Group. *The OpenCL Specification*, March 2014.

[21] S. E. Gryning and E. Lyck. Atmospheric dispersion from elevated sources in an urban area: Comparison between tracer experiments and model calculations. *Journal of Climate and Applied Meteorology*, 23:651 – 660, 1984. ISSN 0733-3021.

[22] S. E. Gryning and E. Lyck. The Copenhagen tracer experiments: Reporting of measurements. Technical report, 1998. 327–366 pp.

[23] S. R. Hanna. Confidence limits for air quality model evaluations, as estimated by bootstrap and jackknife resampling methods. *Atmospheric Environment*, 23:1385–1398, 1989. ISSN 0004-6981.

[24] I. Herrera. *Boundary Methods: An Algebraic Theory*. Pitman Publishing Limited, 1984. ISBN 978-0-273-08635-2.

[25] K. A. Hoffmann and S. T. Chiang. *Computational Fluid Dynamics*, volume 1. EES, $4^{th}$ edition, 2000. ISBN 978-0-962-37310-7.

[26] W. Hundsdorfer and J. G. Verwer. *Numerical Solution of Time-Dependent Advection-Diffusion-Reaction Equations*. Springer-Verlag, 2007. ISBN 978-3-662-09017-6.

[27] C. H. Hunter. A recomended Pasquill-Gifford stability classification method for safety basis atmospheric dispersion modeling at SRS. Technical report, SRNL, 2012.

[28] G. S. Jiang and C. W. Shu. Efficient implementation of weighted ENO schemes. *Journal of Computational Physics*, 126(1):202 – 228, 1996. ISSN 0021-9991.

[29] B. Koren. A robust upwind discretization for advection, diffusion and source terms. In C.B. Vreugdenhil and B. Koren, editors, *Numerical Methods for Advection - Diffusion Problems*, pages 117–138. Friedrich Vieweg & Sohn Verlagsgesellschaft mbH, 1993. ISBN 978-3-528-07645-0.

[30] D. Lanser and J. G Verwer. Analysis of operator splitting for advection–diffusion–reaction problems from air pollution modelling. *Journal of Computational and Applied Mathematics*, 111(1–2):201–216, 1999. ISSN 0377-0427.

[31] R. W. Lewis, P. Nithiarasu, and K. Seetharamu. *Fundamentals of the Finite Element Method for Heat and Fluid Flow*. Wiley, 2004. ISBN 978-0-470-84789-3.

[32] J. S. Lin and L. M. Hildemann. A generalized mathematical scheme to analytically solve the atmospheric diffusion equation with dry deposition. *Atmospheric Environment*, 31: 59–71, 1997. ISSN 1352-2310.

[33] J. Liu. The white paper on ELLAM implementation in C++. Technical report, 2009.

[34] J. Liu, S. Tavener, and H. Chen. ELLAM for resolving the kinematics of two-dimensional resistive magnetohydrodynamic flows. *Journal of Computational Physics*, 227:1372–1386, December 2007. ISSN 0021-9991.

[35] X. D. Liu, S. Osher, and T. Chan. Weighted essentially non-oscillatory schemes. *Journal of Computational Physics*, 115(1):200 – 212, 1994. ISSN 0021-9991.

[36] W. F. Matthew, E. K. Christopher, F. R. Thomas, and T. M. Cass. An ELLAM approximation for advective-dispersive transport with nonlinear sorption. *Advances in Water Resources*, 29(5):657 – 675, 2006. ISSN 0309-1708.

[37] P. Micikevicius. 3D finite difference computation on GPUs using CUDA. In *Proceedings of* $2^{nd}$ *Workshop on General Purpose Processing on Graphics Processing Units*, pages 79–84. ACM, 2009. ISBN 978-1-605-58517-8.

[38] J. Molemaker, J. M. Cohen, S. Patel, and J. Noh. Low viscosity flow simulations for animation. In *Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 9–18. Eurographics Association, 2008. ISBN 978-3-905-67410-1.

[39] F. Molnar Jr, T. Szakaly, R. Meszaros, and I. Lagzi. Air pollution modelling using a graphics processing unit with CUDA. *Computer Physics Communications*, 181(1): 105–112, 2010. ISSN 0010-4655.

[40] T. Neubauer and P. Bastian. On a monotonicity preserving Eulerian–Lagrangian localized adjoint method for advection–diffusion equations. *Advances in Water Resources*, 28(12):1292 – 1309, 2005. ISSN 0309-1708.

[41] F. Pasquill and F. B. Smith. *Atmospheric diffusion*. E. Horwood, 1983. ISBN 978-0-853-12587-7.

[42] K. Rektorys. *Přehled užité matematiky: I [Survey of applicable mathematics: I]*, volume 7. Prometheus, 2000. ISBN 978-8-071-96180-2.

[43] K. Rektorys. *Přehled užité matematiky: II [Survey of applicable mathematics: II]*, volume 7. Prometheus, 2000. ISBN 978-8-071-96181-9.

[44] K. Ridder and C. Mensink. Improved algorithms for advection and vertical diffusion in Aurora. In Carlos Borrego and Guy Schayes, editors, *Air Pollution Modeling and Its Application XV*, pages 395–401. Springer US, 2002. ISBN 978-0-306-47294-7.

[45] T. F. Russell and P. Binning. Oh no, not the wiggles again! A revisit of an old problem and a new approach. In William G. Gray Cass T. Miller, Matthew W. Farthing and George F. Pinder, editors, *Computational Methods in Water Resources: Volume 1*, volume 55, Part 1 of *Developments in Water Science*, pages 483 – 494. Elsevier, 2004. ISSN 0167-5648.

[46] T. F. Russell and M. A. Celia. An overview of research on Eulerian-Lagrangian localized adjoint methods (ellam). *Advances in Water Resources*, 25:1215–1231, 2002. ISSN 0309-1708.

[47] V. Satek. *Analýza stiff soustav diferenciálních rovnic [Stiff Systems Analysis]*. PhD thesis, Department of Intelligent Systems FIT BUT, 2012.

[48] G. A. Sehmel and W. H. Hodgson. A model for predicting dry deposition of particles and gases to environmental surfaces. Technical report, Battelle Pacific Northwest Labs., Richland, WA (USA), 1978.

[49] M. Sharan and S. Gupta. Two-dimensional analytical model for estimating cross-wind integrated concentration in a capping inversion: eddy diffusivity as a function of downwind distance from the source. *Atmospheric Environment*, 36:97–105, 2002. ISSN 1352-2310.

[50] C. W. Shu. Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws. In Alfio Quarteroni, editor, *Advanced Numerical Approximation of Nonlinear Hyperbolic Equations*, volume 1697 of *Lecture Notes in Mathematics*, pages 325–432. Springer Berlin Heidelberg, 1998. ISBN 978-3-540-64977-9.

[51] P. Slanco, E. Geldova, and M. Hanculak, J. Bobro. Teoretický výpočet spadu tuhých častíc z ovzdušia [Theoretical calculation of solid particles deposition from the air]. *Acta Montanistica Slovaca*, 7:79–83, 2002. ISSN 1335-1788.

[52] S. A. Socolofsky and G. H. Jirka. *Environmental Fluid Mechanics 1: Mixing and Transport Processes in the Environment.* Texas A&M University, $5^{th}$ edition, 2005.

[53] G. Strang. On the construction and comparison of difference schemes. *SIAM Journal on Numerical Analysis*, 5(3):506–517, sep 1968. ISSN 1095-7170.

[54] K. A. Tan, R. P. Morison, and L. M. Leslie. A comparison of high-order explicit and non-oscillatory finite difference advection schemes for climate and weather models. *Meteorology and Atmospheric Physics*, 89(1-4):251–267, 2005. ISSN 0177-7971.

[55] J. Tolke and M. Krafczyk. TeraFLOP computing on a desktop PC with GPUs for 3D CFD. *International Journal of Computational Fluid Dynamics*, 22(7):443–456, 2008. ISSN 1061-8562.

[56] A. G. Ulke. New turbulent parameterization for a dispersion model in the atmospheric boundary layer. *Atmospheric Environment*, 34(7):1029–1042, 2000. ISSN 1352-2310.

[57] D. A. Vallero. *Fundamentals of Air Pollution (Fourth Edition)*. Elsevier, 2008. ISBN 978-0-123-73615-4.

[58] C. J. Walcek and N. M. Aleksic. A simple but accurate mass conservative, peak-preserving, mixing ratio bounded advection algorithm with FORTRAN code. *Atmospheric Environment*, 32(22):3863 – 3880, 1998. ISSN 1352-2310.

[59] H. Wang, R. E. Wing, G. Qin, S. L. Lyons, M. Al-Lawatia, and S. Man. A family of Eulerian-Lagrangian localized adjoint methods for multi-dimensional advection-reaction equations. *J. Comput. Phys.*, 152(1):120–163, 1999. ISSN 0021-9991.

[60] S. Wortmann, M. T. Vilhena, D. M. Moreira, and D. Buske. A new analytical approach to simulate the pollutant dispersion in the PBL. *Atmospheric Environment*, 39(12): 2171–2178, 2005. ISSN 1352-2310.

[61] A. Younes and P. Ackerer. Solving the advection-diffusion equation with the Eulerian–Lagrangian localized adjoint method on unstructured meshes and non uniform time stepping. *Journal of Computational Physics*, 208:384–402, 2005. ISSN 0021-9991.

[62] A. Younes, P. Ackerer, and F. Lehmann. A new efficient Eulerian–Lagrangian localized adjoint method for solving the advection–dispersion equation on unstructured meshes. *Advances in Water Resources*, 29:1056—-1074, 2006. ISSN 0309-1708.

[63] A. Younes, M. Fahs, and P. Ackerer. A new approach to avoid excessive numerical diffusion in Eulerian–Lagrangian methods. *Communications in Numerical Methods in Engineering*, 24(11):897–910, 2008. ISSN 1099-0887.

# List of publications related to PhD thesis


[DKZZ10] R. Dvorak, J. Krajicek, J. Zendulka, and F. Zboril. The design of self-developmental model of atmospheric pollutant dispersion. In *Proceedings of the 7th EUROSIM Congress on Modelling and Simulation*, page 6. Czech Technical University Publishing House, 2010. ISBN 978-8-001-04589-3.

[DZ08a] R. Dvorak and F. Zboril. Modeling of atmospheric dispersion from point source. In *Proceedings of the* $2^{nd}$ *UKSim European Symposium on Computer Modelling and Simulation*, pages 46–51. IEEE Computer Society, 2008. ISBN 978-0-769-53325-4.

[DZ08b] R. Dvorak and F. Zboril. Solving of advection-diffusion equation using method of lines. In *Proceedings* $8^{th}$ *International Scientific Conference on Computers Science and Engineering*, pages 305–311. The University of Technology Kosice, 2008. ISBN 978-8-080-86092-9.

[DZ09] R. Dvorak and F. Zboril. Simulation of atmospheric pollution dispersion. In *Proceedings MATHMOD 09 Vienna - Full Papers CD Volume*, pages 2671–2674. ARGE Simulation News, 2009. ISBN 978-3-901-60835-3.

[DZ10] R. Dvorak and F. Zboril. Preliminary study of using Ellam framework for solution of atmospheric advection diffusion reaction equation. In *Proceedings of the* $7^{th}$ *EUROSIM Congress on Modelling and Simulation*, page 6. Czech Technical University Publishing House, 2010. ISBN 978-8-001-04589-3.

[DZ11] R. Dvorak and F. Zboril. On the usage of ELLAM to solve advection-diffusion equation describing the pollutant transport in planetary boundary layer. In *Proceedings of the Eleventh International Conference on Informatics*, pages 227–230. Faculty of Electrical Engineering and Informatics, University of Technology Kosice, 2011. ISBN 978-8-089-28494-8.

[DZ12] R. Dvorak and F. Zboril. On the usage of ELLAM to solve advection-diffusion equation describing the pollutant transport in planetary boundary layer. *Acta Electrotechnica et Informatica*, 12(1):44–48, 2012. ISSN 1335-8243.

[DZS09a] R. Dvorak, F. Zboril, and V. Simek. A numerical solution of the dispersion modeling in the planetary boundary layer. In *2009 International Conference on Computational Intelligence, Modelling and Simulation*, pages 6–10. IEEE Computer Society, 2009. ISBN 978-0-769-53795-5.

[DZS09b] R. Dvorak, F. Zboril, and P. Srnec. Adding a rough terrain to atmospheric pollutant dispersion model. In *Proceedings of the Tenth International Conference of Informatics*, pages 221–226. The University of Technology Kosice, 2009. ISBN 978-8-080-86126-1.

[KDZM08] M. Kapoun, R. Dvorak, F. Zboril, and I. Masek. Modeling of dispersion of windborne material in atmosphere. *Chemické listy*, 102(15):386–391, 2008. ISSN 0009-2770.

[SDZD09] V. Simek, R. Dvorak, F. Zboril, and V. Drabek. GPU accelerated solver of time-dependent air pollutant transport equations. In $12^{th}$ *EUROMICRO Conference on Digital System Design*, pages 707–713. IEEE Computer Society, 2009. ISBN 978-0-769-53277-6.

[SDZD10] V. Simek, R. Dvorak, F. Zboril, and V. Drabek. Performance evaluation of OpenCL framework for numerical solver of advection diffusion equation. In *Proceedings of CSE 2010 International Scientific Conference on Computer Science and Engineering*, pages 279–286. The University of Technology Kosice, 2010. ISBN 978-8-080-86164-3.

[SDZK09] V. Simek, R. Dvorak, F. Zboril, and J. Kunovsky. Towards accelerated computation of atmospheric equations using CUDA. In *Proceedings of Eleventh International Conference on Computer Modelling and Simulation*, pages 449–454. IEEE Computer Society, 2009. ISBN 978-0-769-53593-7.

# List of other publications


[DD07] R. Dvorak and M. Drahansky. Real-time terrain deformations. In *Fourth International Conference on Image and Graphics*, pages 1020–1025. IEEE Computer Society, 2007. ISBN 0-769-52929-1.

[DD08] R. Dvorak and M. Drahansky. Physically based real-time terrain deformations: Extensions. In *Proceedings of* $18^{th}$ *International Conference on Computer Graphics and Vision*, pages 74–78. Lomonosov Moscow State University, 2008. ISBN 978-5-955-60112-0.

[DDO09] R. Dvorak, M. Drahansky, and F. Orsag. Object surface reconstruction from one camera system. In *Future Generation Information Technology*, pages 87–94. Springer, 2009. ISBN 978-3-642-10508-1.

[DDV11a] M. Drahansky, R. Dvorak, and J. Vana. Personenerkennung mittels 3D Handgeometrie. *Datenschutz und Datensicherheit-DuD*, 35(3):169–173, 2011. ISSN 1614-0702.

[DDV11b] R. Dvorak, M. Drahansky, and J. Vana. Human recognition by 3D hand geometry. In *Proceedings of Security and Protection of Information 2011*, pages 169–177. Brno University of Defence, 2011. ISBN 978-8-072-31777-6.

[DDVD12] O. Dluhos, R. Dvorak, J. Vana, and M. Drahansky. Hand surface as biometric identifier. *Advanced Science and Technology Letters*, 8(7):351–356, 2012. ISSN 2287-1233.

[DLD+10] R. Dvorak, D. Lodrova, P. Dittrich, J. Vana, and M. Drahansky. Research in the area of biometric systems - liveness detection; recognition of 3D hand, finger veins and thermofaces. In *World and homeland security*, pages 143–152. Brno University of Defence, 2010. ISBN 978-8-072-31728-8.

[DOea11] M. Drahansky and F. Orsag et al. *Biometrie [Biometrics].* Computer Press, s.r.o, $1^{st}$ edition, 2011. ISBN 978-8-025-48979-6. 294 pp.

[HDDO09] D. Hejtmankova, R. Dvorak, M. Drahansky, and F. Orsag. A new method of finger veins detection. *International Journal of Bio-Science and Bio-Technology*, 1(1):11–15, 2009. ISSN 2233-7849.

[LDDO09] D. Lodrova, R. Dvorak, M. Drahansky, and F. Orsag. A new approach for veins detection. In *Bio-Science and Bio-Technology*, pages 76–80. Springer Berlin Heidelberg, 2009. ISBN 978-3-642-10615-6.

[MBDD11] S. Mracek, Ch. Busch, R. Dvorak, and M. Drahansky. Inspired by Bertillon - recognition based on anatomical features from 3D face scans. In *Proceedings of the $3^{rd}$ International Workshop on Security and Communication Networks*, pages 53–58. Gjovik University College, 2011. ISBN 978-8-291-31367-2.

[MHV+10] A. Marvan, J. Hajek, J. Vana, R. Dvorak, M. Drahansky, R. Jankovych, and J. Skvarek. Scanning and measuring device for diagnostic of barrel bore. In *Signal Processing and Multimedia*, pages 24–29. Springer, 2010. ISBN 978-3-642-17640-1.

[MVD+12] S. Mracek, J. Vana, R. Dvorak, M. Drahansky, and S. Yanushkevich. 3D and thermo-face fusion. In *New Trends and Developments in Biometrics*, pages 31–58. InTech - Open Access Publisher, 2012. ISBN 978-9-535-10859-7.

[MVY+12] S. Mracek, J. Vana, S. Yanushkevich, M. Drahansky, and R. Dvorak. Thermal face recognition: a fusion approach. In *Proceedings of the Emerging Security Technologies*, pages 39–42. IEEE Computer Society, 2012. ISBN 978-0-769-54791-6.

[VDD11] J. Vana, M. Drahansky, and R. Dvorak. Thermal face recognition - matching algorithms performance. In *Proceedings of the Conference Security and Protection of Information 2011*, pages 140–149. Brno University of Defence, 2011. ISBN 978-8-072-31777-6.

# Appendix A

# MoL/Stability assumptions - derivation of equation with amplification factor

This appendix contains the derivation of equation (4.29) which is in the form containing the amplification factor. The equation and identities at the beginning of the derivation are repeated here for clarity (these are (A.1), (A.2) and (A.3)). The initial equation has the following form

$$\begin{aligned}\frac{\partial C(t,i,j,k)}{\partial t} = &-\frac{a_x}{2\Delta x}\left[C(t,i+1,j,k) - C(t,i-1,j,k)\right] + \\ &\frac{D_y(x)}{\Delta y^2}\left[C(t,i,j+1,k) - 2C(t,i,j,k) + C(t,i,j-1,k)\right] + \\ &\frac{D_z(x)}{\Delta z^2}\left[C(t,i,j,k+1) - 2C(t,i,j,k) + C(t,i,j,k-1)\right] + \\ &\frac{W}{2\Delta z}\left[C(t,i,j,k+1) - C(t,i,j,k-1)\right].\end{aligned} \tag{A.1}$$

The transformation to phase space is defined as

$$C(t,i,j,k) = U(t)e^{I(\Theta i+\Phi j+\Psi k)}. \tag{A.2}$$

The advection coefficients, $\alpha_a$ and $\alpha_W$, and diffusion coefficients, $\beta_y$ and $\beta_z$ are defined as

$$\begin{aligned}\alpha_a &= \frac{a_x\Delta t}{2\Delta x}, \quad \alpha_W = \frac{W\Delta t}{2\Delta z}, \\ \beta_y &= \frac{D_y\Delta t}{(\Delta y)^2}, \quad \beta_z = \frac{D_z\Delta t}{(\Delta z)^2}.\end{aligned} \tag{A.3}$$

After substituting terms (A.3) into equation (A.1) and applying transformation (A.2) the following form is obtained

$$
\begin{aligned}
U(t+\Delta t)e^{I(\Theta i+\Phi j+\Psi k)} = &-\alpha_a \left[U(t)e^{I(\Theta(i+1)+\Phi j+\Psi k)} - U(t)e^{I(\Theta(i-1)+\Phi j+\Psi k)}\right] + \\
&\beta_y \left[U(t)e^{I(\Theta i+\Phi(j+1)+\Psi k)} - 2U(t)e^{I(\Theta i+\Phi j+\Psi k)} + U(t)e^{I(\Theta i+\Phi(j-1)+\Psi k)}\right] + \\
&\beta_z \left[U(t)e^{I(\Theta i+\Phi j+\Psi(k+1))} - 2U(t)e^{I(\Theta i+\Phi j+\Psi k)} + U(t)e^{I(\Theta i+\Phi j+\Psi(k-1))}\right] + \\
&\alpha_W \left[U(t)e^{I(\Theta i+\Phi j+\Psi(k+1))} - U(t)e^{I(\Theta i+\Phi j+\Psi(k-1))}\right].
\end{aligned} \tag{A.4}
$$

After dividing by $e^{I(\Theta i+\Phi j+\Psi k)}$, the equation (A.4) leads to

$$
\begin{aligned}
U(t+\Delta t) = &-\alpha_a \left[U(t)e^{I\Theta} - U(t)e^{-I\Theta}\right] + \\
&\beta_y \left[U(t)e^{I\Phi} - 2U(t) + U(t)e^{-I\Phi}\right] + \\
&\beta_z \left[U(t)e^{I\Psi} - 2U(t) + U(t)e^{-I\Psi}\right] + \\
&\alpha_W \left[U(t)e^{I\Psi} - U(t)e^{-I\Psi}\right].
\end{aligned} \tag{A.5}
$$

The right side of equation (A.5) can be further simplified into

$$
\begin{aligned}
U(t+\Delta t) = U(t)\{&-\alpha_a \left[e^{I\Theta} - e^{-I\Theta}\right] + \\
&\beta_y \left[-2 + e^{I\Phi} + e^{-I\Phi}\right] + \\
&\beta_z \left[-2 + e^{I\Psi} + e^{-I\Psi}\right] + \\
&\alpha_W \left[e^{I\Psi} - e^{-I\Psi}\right]\}.
\end{aligned} \tag{A.6}
$$

Applying the following identities [42]

$$
\sin\alpha = \frac{e^{I\alpha} - e^{-I\alpha}}{2I}, \tag{A.7}
$$

$$
\cos\alpha = \frac{e^{I\alpha} + e^{-I\alpha}}{2}, \tag{A.8}
$$

to equation (A.6) results in

$$
\begin{aligned}
&U(t+\Delta t) = \\
&U(t)\left[-\alpha_a 2I\sin\Theta + \beta_y\left(-2+2\cos\Phi\right) + \beta_z\left(-2+2\cos\Psi\right) + \alpha_W 2I\sin\Psi\right].
\end{aligned} \tag{A.9}
$$

The rearranging of terms in equation (A.9) leads to the final form with the amplification factor $G$

$$
\begin{aligned}
U(t+\Delta t) =&[1 + 2\beta_y(\cos\Phi - 1) + 2\beta_z(cos\Psi - 1) \\
&- I(2\alpha_a\sin\Theta - 2\alpha_W\sin\Psi)] \cdot U(t) \\
=&G \cdot U(t).
\end{aligned} \tag{A.10}
$$

# Appendix B

# Detailed experiment images

Some detailed images from advection experiments of ELLAM and the Adapt. ELLAM are presented in this appendix.

## B.1 Rotation wind

The details at the bottom and at the top of the concentration shapes for all rotated experiments (60 rotations) are shown here.

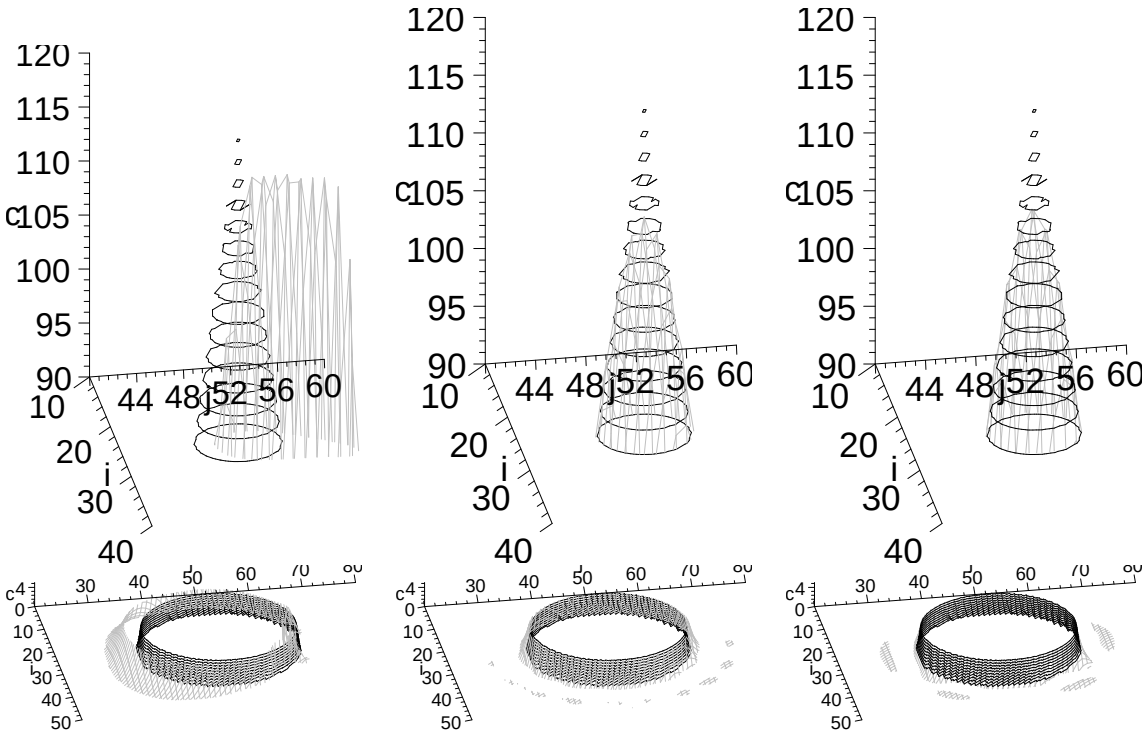



Figure B.1: The comparison of top and bottom parts of the final concentration shapes (cone) with the exact solution: Walcek (left), ELLAM (center) and Adapt. ELLAM (right).

It could be seen from the figures that the concentration shapes are a bit deformed in case of Walcek method. On the other hand, the oscillations are visible in the images of ELLAM method and their reduction in the images of Adapt. ELLAM method.

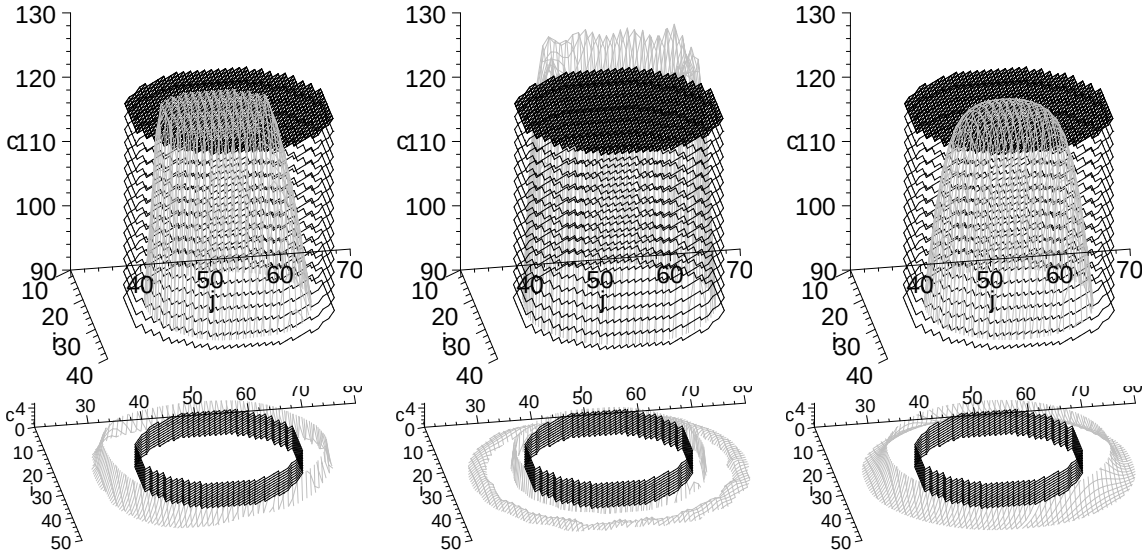


Figure B.2: The comparison of top and bottom parts of the final concentration shapes (cylinder) with the exact solution: Walcek (left), ELLAM (center) and Adapt. ELLAM (right).

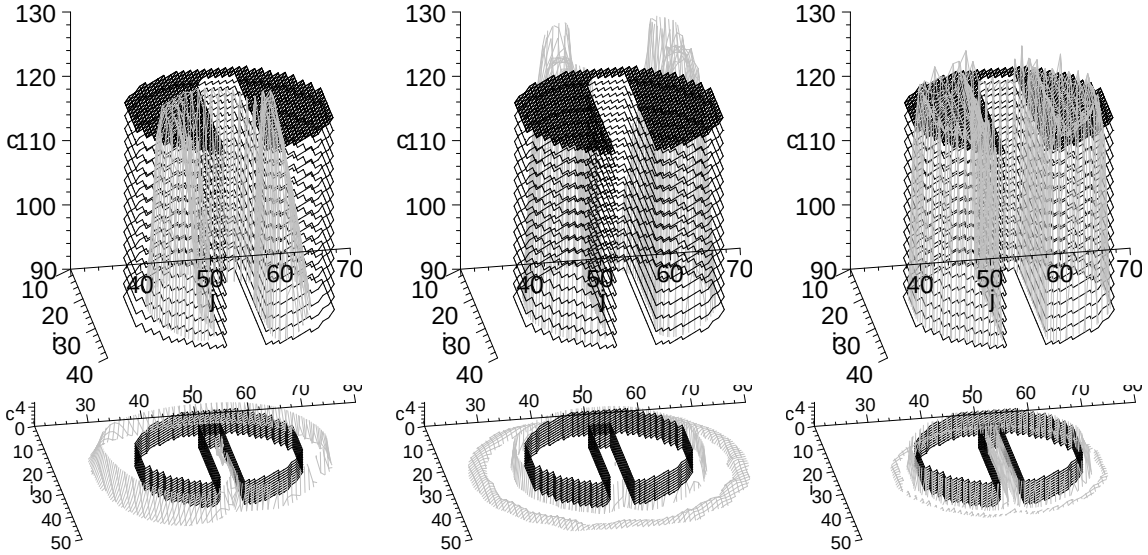


Figure B.3: The comparison of top and bottom parts of the final concentration shapes (cylinder) with the exact solution: Walcek (left), ELLAM (center) and Adapt. ELLAM (right).

## B.2 Divergent wind

The evaluation of the concentration shape profile in case of divergent wind is shown here. Walcek and Adapt. ELLAM cases are shown. It is shown that in case of Adapt. ELLAM, the tracer was out-flowed to surrounding wind squares due to small oscillations and the added diffusion.

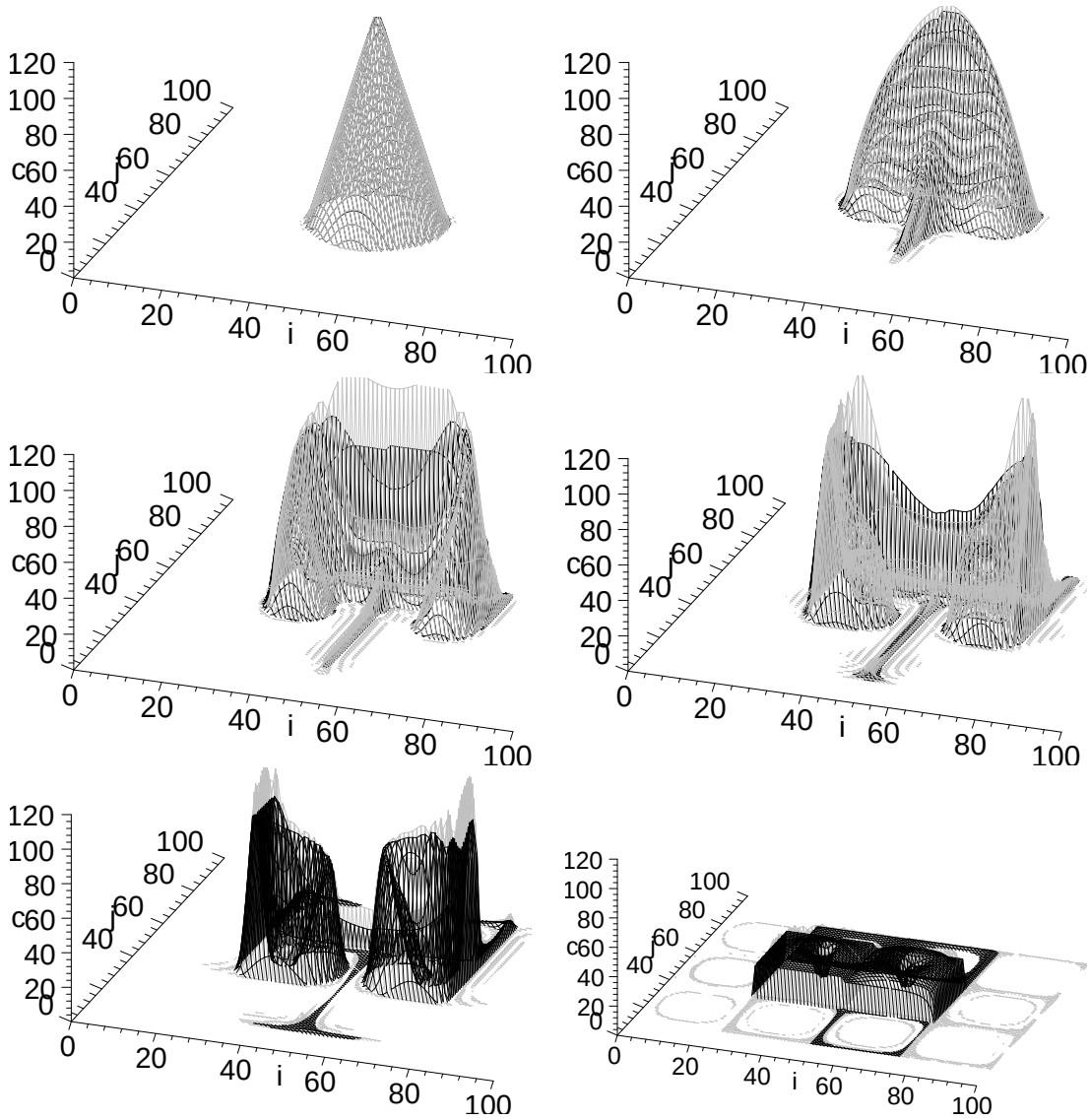



Figure B.4: The results after 1, 20, 40, 60, 80 and 4000 Walcek steps (step size was set to 2637.6 seconds) for Walcek (black) and Adapt. ELLAM (grey) methods.